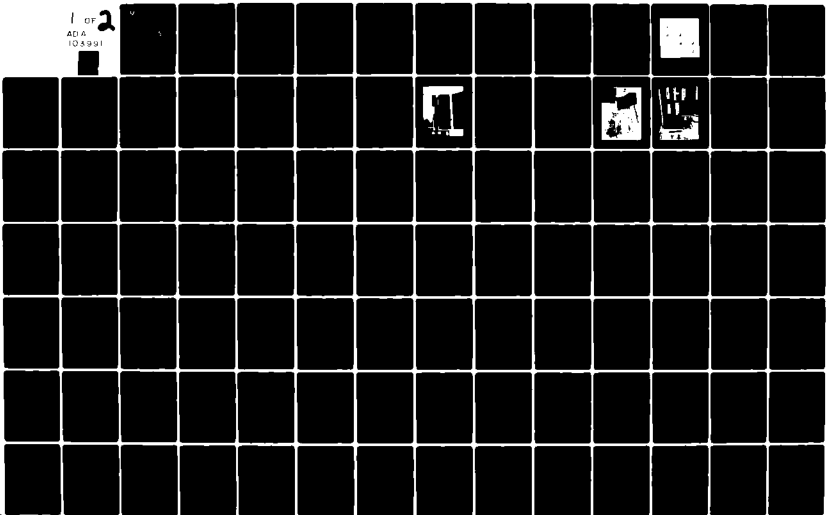
AD-A103 991 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND FO--ETC F/G 9/2
TACTICAL SITUATION SIMULATOR ALGORITHM FOR USE WITH A THERMAL L--ETC(U)
AUG 81 J A KARAKOWSKI, R J MARTINO, A SLUTSKY
UNCLASSIFIED DELCS-TR-81-1

1 OF 2
AD A
103991



AD A103991

LEVEL II

4

RESEARCH AND DEVELOPMENT TECHNICAL REPORT

DELCS-TR-81-1

TACTICAL SITUATION SIMULATOR ALGORITHM FOR USE WITH A THERMAL LINE PRINTER IN A SENSOR MONITORING SET

J. KARAKOWSKI
R. MARTINO
A. SLUTSKY

COMBAT SURVEILLANCE & TARGET ACQUISITION LABORATORY

AUGUST 1981

DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

ERADCOM

US ARMY ELECTRONICS RESEARCH & DEVELOPMENT COMMAND
FORT MONMOUTH, NEW JERSEY 07703

81 9 08 001

NOTICES

Disclaimers

The citation of trade names and names of manufacturers in this report is not to be construed as official Government indorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 14 DELCS-TR-81-1	2. GOVT ACCESSION NO. AD-A103	3. RECIPIENT'S CATALOG NUMBER 997
4. TITLE (and Subtitle) 6 Tactical Situation Simulator Algorithm for use with a Thermal Line Printer in a Sensor Monitoring Set.		5. TYPE OF REPORT & PERIOD COVERED 9 Final rept.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 J.A. Karakowski R.J. Martino A. Slutsky		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS USA ERADCOM DELCS-I Ft. Monmouth, NJ 07703		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS N/A 12 117
11. CONTROLLING OFFICE NAME AND ADDRESS Combat Surveillance and Target Acquisition ATTN: DELCS-I Ft. Monmouth, NJ 07703		12. REPORT DATE 11 August 1981
		13. NUMBER OF PAGES 116
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approval for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Tactical Simulator, Unattended Sensors, Computer Programs, Thermal Recorder, Displays, Real Time Simulator, Human Factors.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A simulator which generated unattended ground sensor activations and target classification was developed and used in conjunction with a Thermal Recorder Display for a limited Human Factor Test to determine the usefulness of classification symbols in operator processing of sensor data.

The resulting data indicates that these symbols do aid in classification and object discrimination.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 387

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

APPENDICES

FIGURES

TABLES

DEVELOPMENT OF TACTICAL SITUATION SIMULATOR ALGORITHM AND INVESTIGATION OF THERMAL LINE PRINTER FOR A SENSOR MONITORING SET

INTRODUCTION

A Sensor Monitoring Set (SMS) is being developed to monitor unattended group sensors. This device displays sensor alarms on an X-T recorder which presents an operator with a time history of sensor activations and target classification data. This data and the resulting activation patterns generated can be used to calculate and determine target parameters such as direction, velocity, length to column, number of objects in a target, et cetera.

Efficient processing of this data by an operator is partly dependent on the manner in which the data is presented. This area is presently under investigation. Included in this investigation is a limited human factors test which was conducted using Army and Marine personnel at Ft. Monmouth, NJ. However, owing to constraints imposed by funding and time limitations, it was recognized that the scope of the effort would be limited to a modest investigation that, at best, would produce only indicators on formats for displaying sensor data and on the performance of operators with visual display formats.

In order to display actual sensor activations and target classification for these tests, a data base containing these activations was required. A tactical situation simulator was designed to approximate operational situations and generate the resulting activations to be displayed.

The system and tests this report discusses were designed to aid in this investigation.

GENERAL DESCRIPTION

The system utilized an Interdata Model 70 minicomputer with peripheral devices, a tactical situation simulator, character generator, and associated recorder programs. Generated sensor activations are processed to determine sensor type and target classification, if applicable. This processed data is displayed on the recorder for operator processing. Additional displays are used by the personnel conducting the tests to monitor the simulator outputs.

Human factors tests were conducted using trained operators who were asked to extract as much target parameter information as they could from the X-T plots. These plots represented a time history of the sensor activations and target classifications which could be expected from various operational tactical situations. Each tactical situation consisted of various types and quantities of vehicles and personnel moving along a number of different trails. Target classifications received from the "sensors" were printed on the plots using symbols and alphabetic characters.

TACTICAL SITUATION SIMULATOR ALGORITHM (TSSA)

The TSSA reproduces the real time response of an unattended ground sensor or group of sensors for any set of objects following a defined set of tracks. Each item - sensors, objects, and road - can be defined by the user and are limited in number only by computer memory size and system cycle speeds. For example, situations which contain upwards of 50 objects and 45 sensors have been used to date, with still larger situations possible. An example will be given later.

The objects are output to an alphanumeric-type CRT in a quasi-graphical mode, that is, individual symbolic object data is mapped in the discrete position on the CRT which is nearest the exact object position. The activations data is output to the thermal recorder, line printer, and CRT. An example of a complete CRT mapping of both object and sensor data is shown in Figure 1. Here, the sensor ID numbers and activation count are given for each sensor and the real time track of the object(s) is displayed.

Models

The geometry, object and sensor functions are described by models of their respective operations. Each of these models was chosen to provide algorithm flexibility as well as a realistic representation of actual system operation. Within the object and sensor models there are also submodels to give further system flexibility.

1. Track Geometry Model

The tracks or trails which an object follows may be a straight line or an approximate curve. The only restriction is that they all lie in the same geometrical plane. Thus, hills or valleys are not accounted for in this model, although they can be implemented with some object restrictions or additional software.

The tracks are defined as piecewise linear approximation in a two-step process. First, a number of discrete straight line segments are defined. Then, these segments are joined together in strings to form the desired tracks.

Each segment is defined in absolute terms by its beginning and end coordinates. When segments are joined, the end coordinates of one must

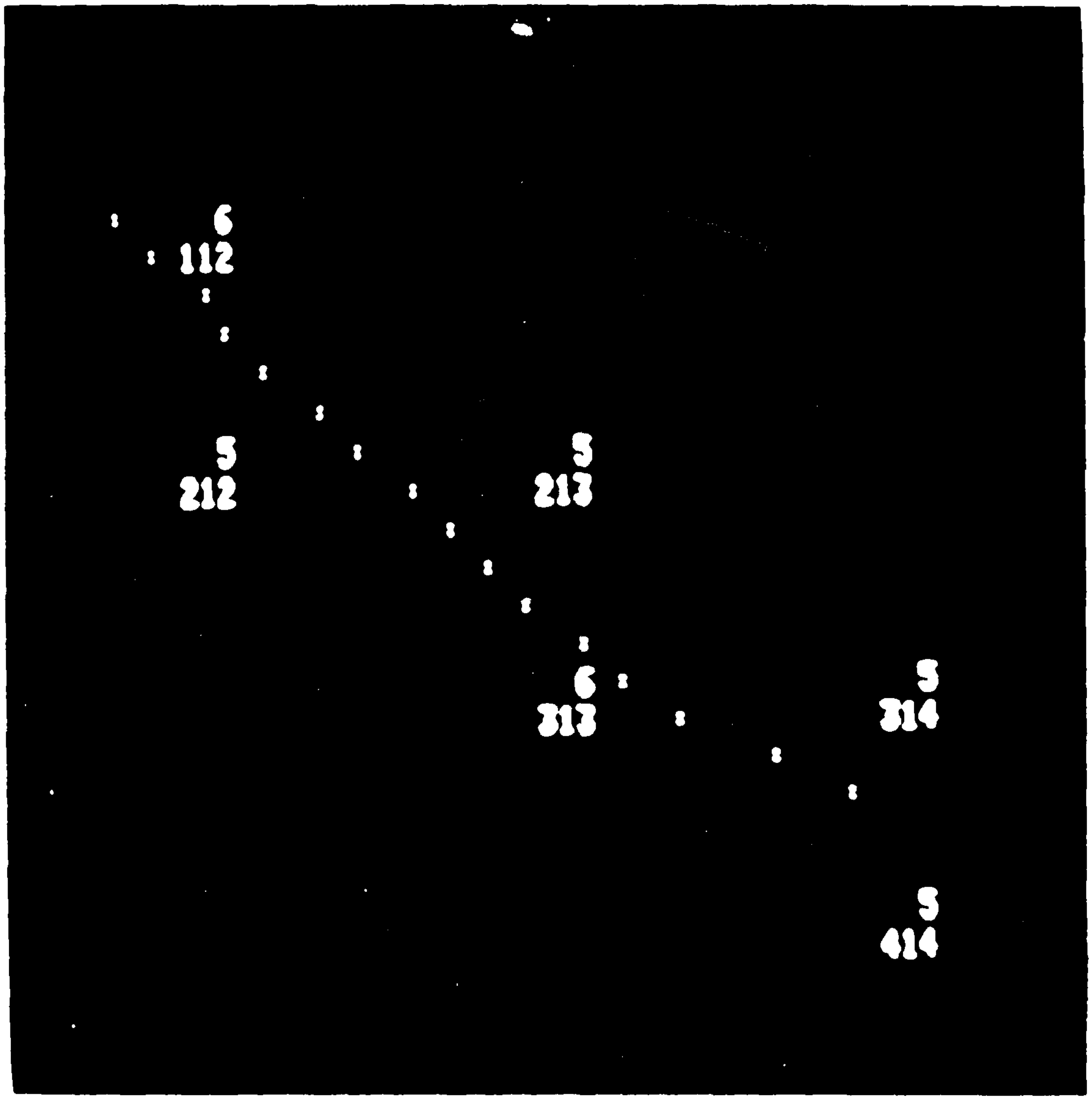


FIGURE 1 REAL TIME CRT TARGET TRACK

equal the start of the next and so on, for useful results to occur. The strings thus formed are numbered to allow assignment to a specific object.

2. Object Model

The objects which are presently recognized by the algorithm are personnel, wheel and track, but this can be expanded up to 255. Each object has several parameters which define its operational characteristics: (1) type, (2) string (track), (3) speed, (4) direction, (5) start time, and (6) location.

3. Sensor Model

There are two types of sensor models used in the algorithm at present; detection-only and classifiers. The detection-only type provides an alarm indication whereas the classification type provides a classification of the target as determined by the classifier model. Other types can be added, if desired, by simple software modifications.

Modeling of all the sensors operation is done by making the following assumptions and parameterization of sensor operation: (1) each sensor exhibits a probability of detection vs. object distance (usually a circular radius of detection is used); (2) the object type has an effect on the sensor detection radius or detection characteristic; (3) an inhibit time of operation is associated with each sensor.

The classification process also has certain operational characteristics which are incorporated into its model: (1) the sensors classify correctly on a gross probability basis, that is, the overall percentage of correct classifications is given; (2) the heaviest target within the area of influence of a classification sensor is always taken as the dominant target; (3) each classification sensor outputs an alert activation when the target is just outside the sensor's detection zone.

The actual classification is performed on a set partitioning scheme. A random number generator develops a random number between 1 and 32767. This set is partitioned at K*32767 where K is the gross probability of correct classification. Thus, any random number which occurs between 1 and K*32767 will correspond to the dominant target and all others will be considered a false alarm. Note that the random number generator produces the same sequence of random numbers for a certain starting value. The starting value for the random number generator can be any number between 1 and 32767. This number initializes the random sequence and produces an entirely different sequence for each different value. Thus, the classification sequences can be kept constant or varied, if desired, by manipulation of the starting value.

Data Structure

The data structure of the algorithm is important because of the flexibility it allows in programming many tactical situations. It consists of three inputs: sensor parameters, track geometry parameters, and object parameters. Each is independent of the others and can thus be modified individually.

TSSA Operation Details

The actual execution of the algorithm must first be preceded by a preliminary analysis and specification of the desired tactical situation. Note that once a tactical situation has been defined, it does not have to be defined again. The situation layout may be obtained from actual military maps or can be composed in any arbitrary manner, depending on the desired geometry one wishes to use. It is best to fix one or more of the data groups, track geometry, objects, or sensors to minimize confusion and to simplify operation. The two most useful groupings to keep fixed are: (1) geometry or (2) geometry and sensors. With a proper choice of track definitions in planning tactical situations a group of objects can be used with any tactical situation desired.

A complete list of the data for definitions of each parameter is presented below. Each of these data groups is translated to tape for a hard copy of the data. Thus, only a few numbered tapes, which contain all the defining data, can be used over and over again to produce a large variety of tactical situations.

Group parameters:

(1) <u>Track Geometry</u>

(A) Segment definition: inputs numbered in order

1. Initial (X, Y) coordinates in meters

2. Final (X, Y) coordinates in meters

(B) String Definition

1. String Number (ID)

2. List of Consecutive Segment Numbers

(2) <u>Sensors</u> - Data required for each sensor:

(A) Type - detection-only, classifier

(B) (X, Y) Location in meters

(C) Inhibit time in seconds

(D) Detection radius in meters

(E) Probability Table number

(F) ID number - RID

(3) Objects - Data required for each object

(A) Type - personnel, track, wheel

(B) Speed in meters per second

(C) String number

(D) Direction along string

(E) Start time of object

A example of one tactical situation used during the thermal printer test is given in Figures 2 through 6. As can be seen, a large variety of track options were available for different objects to follow. The segmentation of the field is shown in Figure 2, and the actual track (string) definitions are shown in Figure 3. The objects (Tape Nos. 42 and 49, Figures 4 and 7) consist of six columns of varying mixtures of wheel and track vehicles with different start times for a total of 35 objects in all. This particular tactical situation took approximately 40 minutes to run. Figures 5 and 6 define the sensor IDs and deployment which was used for the test.

OPERATOR TEST DESCRIPTION

Two groups of trained operators (total of eight) were used during the test. Each individual received as much personal training and demonstration of the equipment as needed. This involved a description of system operation, both of the tactical situation simulator and thermal recorder, as well as detailed data concerning the sensor field such as detection radius, map of sensor placement along trails, and classification characters. Each operator was given a simple test tactical situation, if desired, for practice and additional instruction.

There were several facts and instructions given prior to beginning the testing. The physical layout and characteristics of the sensors and the target trails were explained. The information was presented on a scaled map of the sensor field and the target trails upon which the distances between sensors and strings were given. The map was posted near the operator, to his right, for immediate visual reference. Since there were no reference marks on the Thermal Line Printer (TLP) paper, the operators were given a simple procedure to calculate time on the TLP output. They were told that every inch of chart paper represented 2 minutes of time passage. This was based on the relationship between chart distance and elapsed time for a chart speed of 30 inches/hour. For example, 1.5 inches of chart movement represents a time passage of 3 minutes. In addition, the operators were told that they could also calculate time by using the 10 second inhibit time between sensor activations. For example, a string of five consecutive sensor activations represented a total time of 50 seconds.

For each sensor, the sensor pen patching information was placed above its respective pen. Also, the detection radius for the sensors was fixed on the recorder face. See Figure 8 for the actual arrangement.

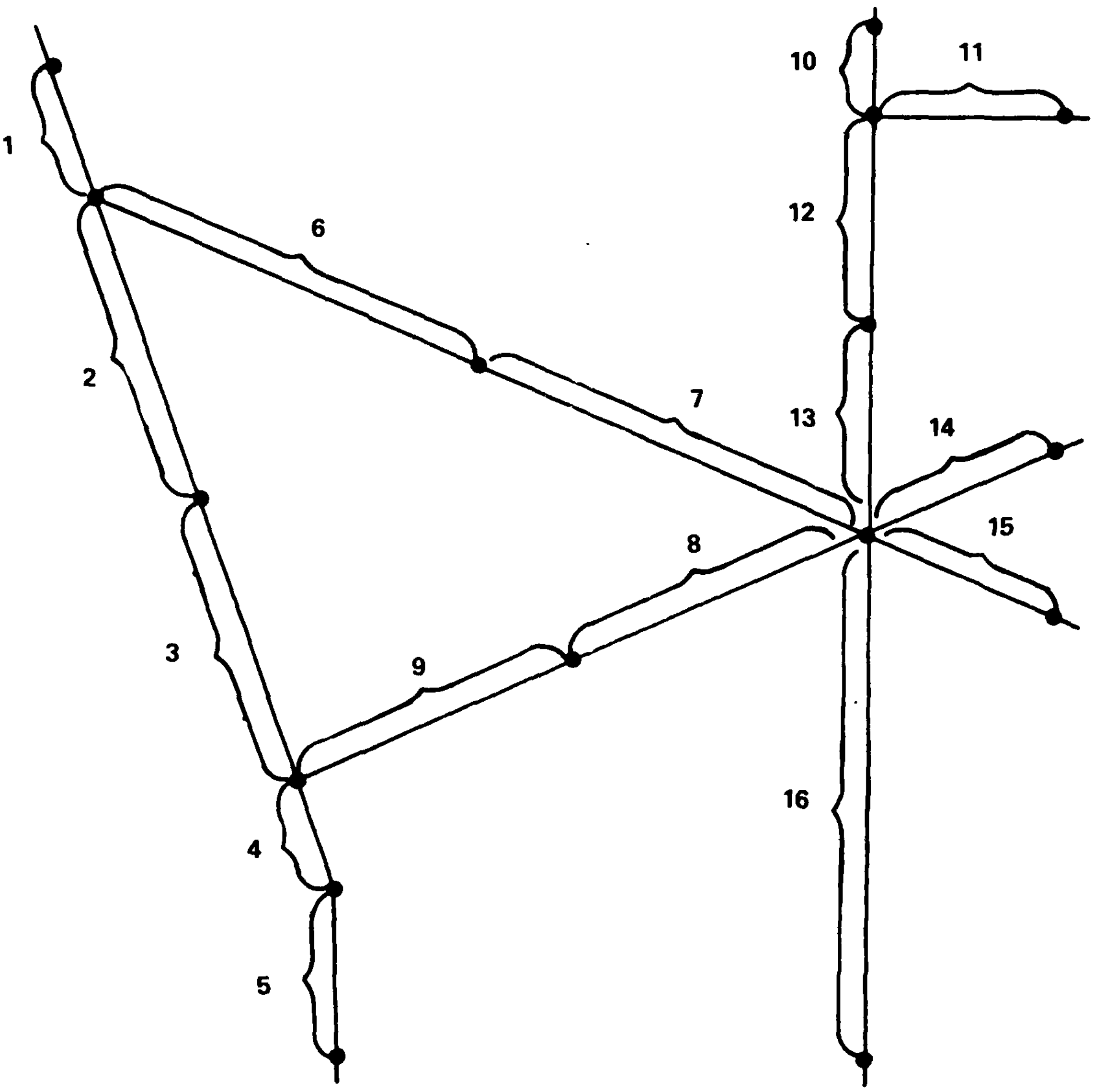


FIGURE 2. TRAIL LAYOUT WITH SEGMENT DEFINITIONS

String #	Segments
1	1, 2, 3, 4, 5
2	3, 4, 5
3	9, 4, 5
4	1, 6, 7, 16
5	7, 16
6	10, 12, 13, 16
7	11, 12, 13, 16
8	13, 16
9	14, 8, 9, 4, 5
10	15, 8, 9, 4, 5
11	10, 12, 13, 8, 9, 4, 5
12	11, 12, 13, 8, 9, 4, 5
13	14, 16
14	15, 16
15	5, 4, 3, 2, 1
16	16, 13, 12, 10
17	8, 14
18	1, 2

FIGURE 3. STRING DEFINITIONS

Object(s)	Type	String	Velocity	Start Time
1	Track	9	10	0:00
2	Track	9	10	0:10
3	Track	9	10	0:20
4	Wheel	9	10	0:30
5	Wheel	9	10	0:40
6	Wheel	9	10	0:50
7-9	Track	6	10	1:55
10	Track	1	10	3:22
11	Track	1	10	3:32
12	Track	1	10	3:42
13	Track	1	10	3:52
14	Wheel	1	10	4:02
15	Wheel	1	10	4:12
16	Wheel	1	10	4:22
17	Track	13	10	12:50
18	Track	13	10	13:00
19	Track	13	10	13:10
20	Track	13	10	13:20
21	Track	13	10	13:30
22	Wheel	13	10	13:40
23	Wheel	13	10	13:50
24	Wheel	13	10	14:00
25	Wheel	13	10	14:10
26	Wheel	13	10	14:20
27	Wheel	13	10	14:30
28-32	Wheel	1	12	13:13
33-35	Track	18	7	17:13

FIGURE 4. TACTICAL SITUATION OBJECT TAPE NO. 42

RID NO.	RECORDER PEN NO.	SENSOR NO.	UTM COORDINATES IN METERS		DISTANCE IN METERS	COMMENT
			EAST	NORTH		
101	101	1	4,389.3	28,832	1,231.1	
102	102	* 2	4,500	28,500	350	
103	103	3	4,657.6	28,132.3	* 400	(S [2-7] = 23.20°)
104	104	4	4,893.9	27,580.9	* 1,000	" "
105	105	5	5,065.6	28,200.6	* 640	(S [2-20] = 62.10°)
206	110	6	7,322.7	21,913.6	6,165.8	*(from S#4)*
207	111	* 7	7,500	21,500	450	(S [2-7] = 23.20°)
208	112	8	7,708.8	21,012.9	* 530	
209	113	9	7,953.0	20,443.0	* 1,150	
211	114	10	8,000.7	21,727.6	* 550	(S [7-20] = 24.44°)
311	117	11	9,000	16,500	4,157.9	(S#9)
312	118	12	9,000	16,100	400	
313	119	13	9,000	15,750	350	
414	122	14	13,000	28,400	1,600	
415	123	* 15	13,000	28,000	400	
416	124	16	13,000	27,500	* 500	
417	125	17	13,000	26,900	* 1,100	
418	126	18	13,600	28,000	* 600	
519	145	19	13,000	24,600	2,300	(S#17)
112	146	* 20	13,000	24,000	600	
521	147	21	13,000	23,400	* 600	
522	148	22	13,000	22,860	* 1,140	
523	149	23	13,409.7	24,186.2	* 450	(S [7-20] = 24.44°)
524	150	24	12,499.3	23,772.4	* 550	" "
525	151	25	11,971.3	23,532.4	* 1,130	" "
526	152	26	12,558.1	24,233.9	* 500	
527	153	27	13,530.3	23,719.3	* 600	
628	155	28	13,000	18,000	4,860	(S#22)
629	156	29	13,000	17,500	500	
113	157	30	13,000	17,050	450	

FIGURE 5. SENSOR DEPLOYMENT CHART

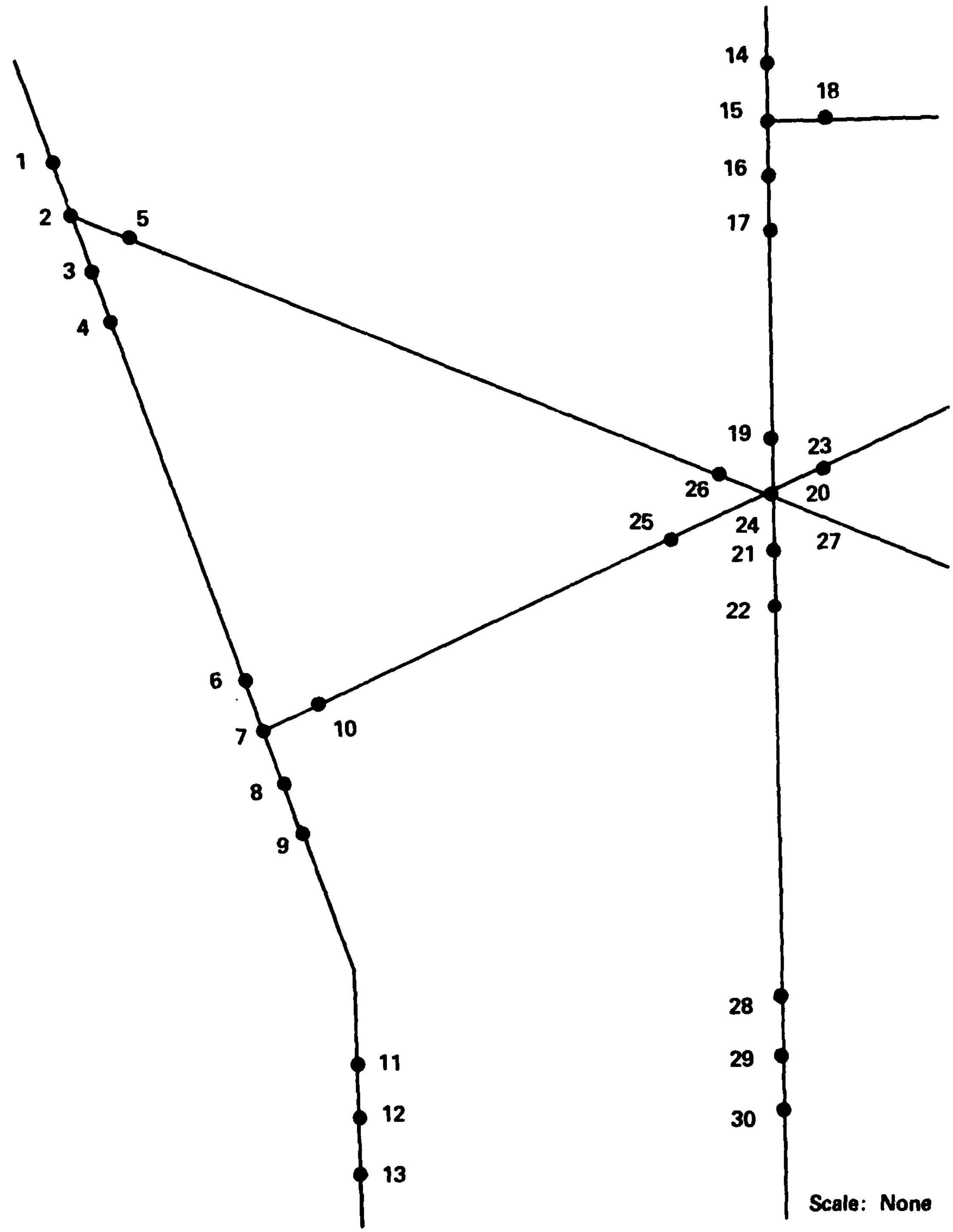


FIGURE 6. SENSOR DEPLOYMENT MAP

Object(s)	Type	String	Velocity	Start Time
1-4	Track	3	10	0:00
5	Track	5	10	0:00
6	Track	5	10	0:10
7	Track	5	10	0:20
8	Wheel	5	10	0:30
9	Wheel	5	10	0:40
10	Track	1	10	2:00
11	Track	1	10	2:10
12	Track	1	10	2:20
13	Track	1	10	2:30
14	Track	1	10	2:40
15	Track	1	10	2:50
16	Track	1	10	3:00
17	Wheel	1	10	3:10
18-26	Personnel	2	1	4:00
27	Track	6	10	4:00
28	Wheel	6	10	4:10
29	Wheel	6	10	4:20
30	Track	6	10	4:30
31	Wheel	6	10	4:40
32	Wheel	6	10	4:50
33	Track	7	10	9:00
34	Wheel	7	10	9:10
35	Wheel	7	10	9:20
36	Wheel	7	10	9:30
37	Wheel	7	10	9:40
38	Wheel	7	10	9:50

FIGURE 7. TACTICAL SITUATION OBJECT TAPE NO. 49

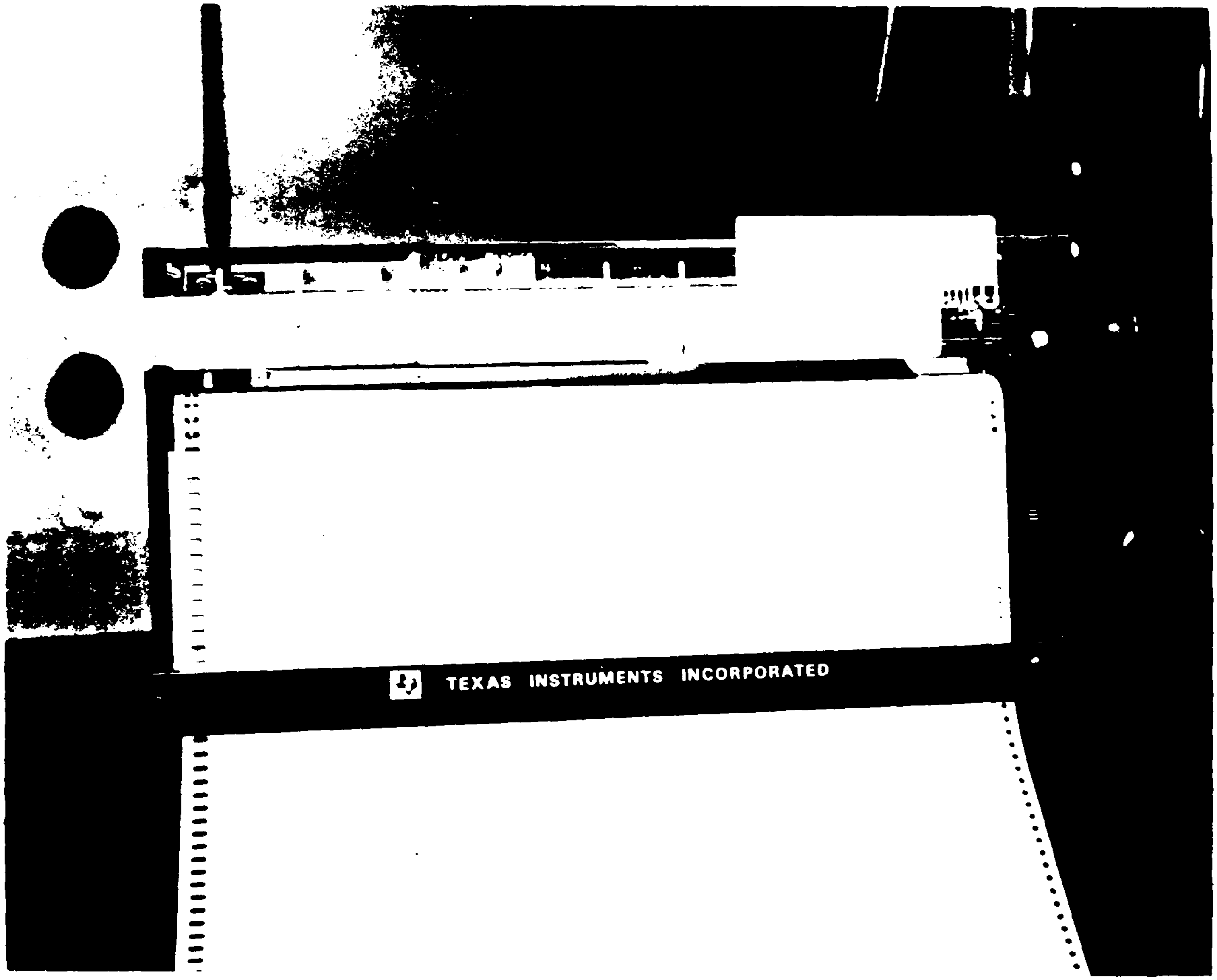


FIGURE 8. TEXAS INSTRUMENTS THERMAL RECORDER

In addition, the operators were:

a. informed that the sensors would classify correctly 80% of the time; that is, 80% of the character symbols displayed on the recorder represented a correct target classification;

b. given a copy and an explanation of the character/symbol set that was to be used for classification sensors;

c. told that they could make use of the classifications to calculate dominant object type and to individually count and classify mixed objects in a target if they desired. They could use any method of analysis available in determining velocity, number of objects in a column, their individual classifications, and direction. Thus, it was left to each operator's discretion to use or ignore the classification data reported on the TLP;

d. told to make all calculations on scratch paper and provide only their results on the actual TLP output.

The layout for each test was as shown in Figure 9. A silent observer was present to note operator comments and performance during the test, as well as to monitor the CRT map as a guide for the tactical situations' progress. The operator received no help from the observer during the progress of each test, except for clarification of sensor data (detection, radius, layout, etc.). There was an audio indication available to each operator, if desired, for each sensor activation. This was provided by the line printer carriage movement as it printed the activation data. Figures 10 and 11 are additional photographs of the operator test area, except for the position of the CRT.

When each test was complete (in the actual output of activations), the operator was notified of the fact by the observer and given time to complete his analysis of the activation data.

The data obtained from each operator consisted of the actual thermal recorder output with the operator's results printed next to each alarm pattern. Also, through personal discussion with the operators, various operational and display techniques which might aid the operator were brought out.

There were several characters/symbols used for display of this test. Each was used with one or more operators. They may be seen in Figure 12 and are cross referenced to the operator test data.

This test was not meant to be a full blown human factors evaluation of the equipment, it should rather be considered a probe.

Operator Calculations

The operators were directed to make calculations of target velocity, target direction, number of objects with a given target, and their classification using any means at their disposal to reach such conclusions. The

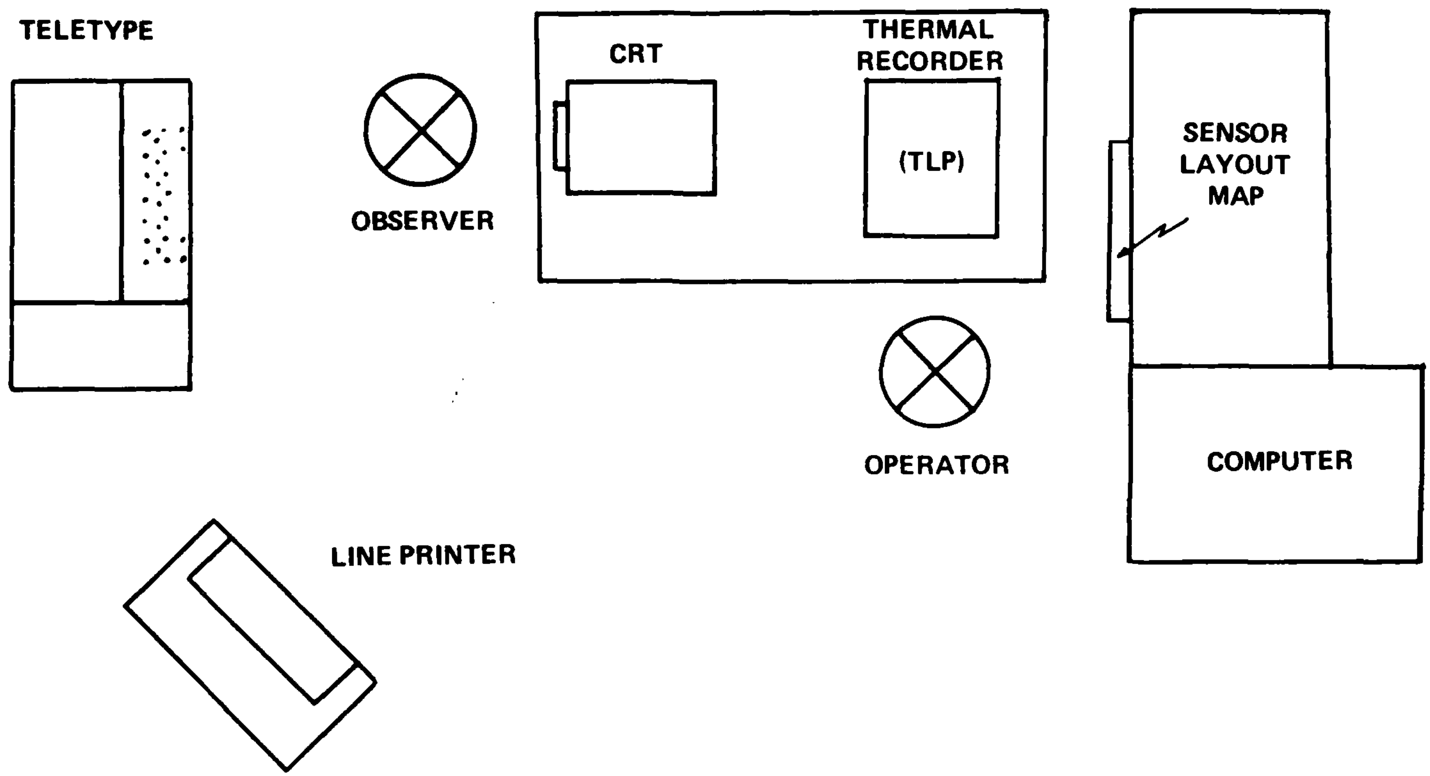


FIGURE 9. OBSERVER—OPERATOR TEST LAYOUT

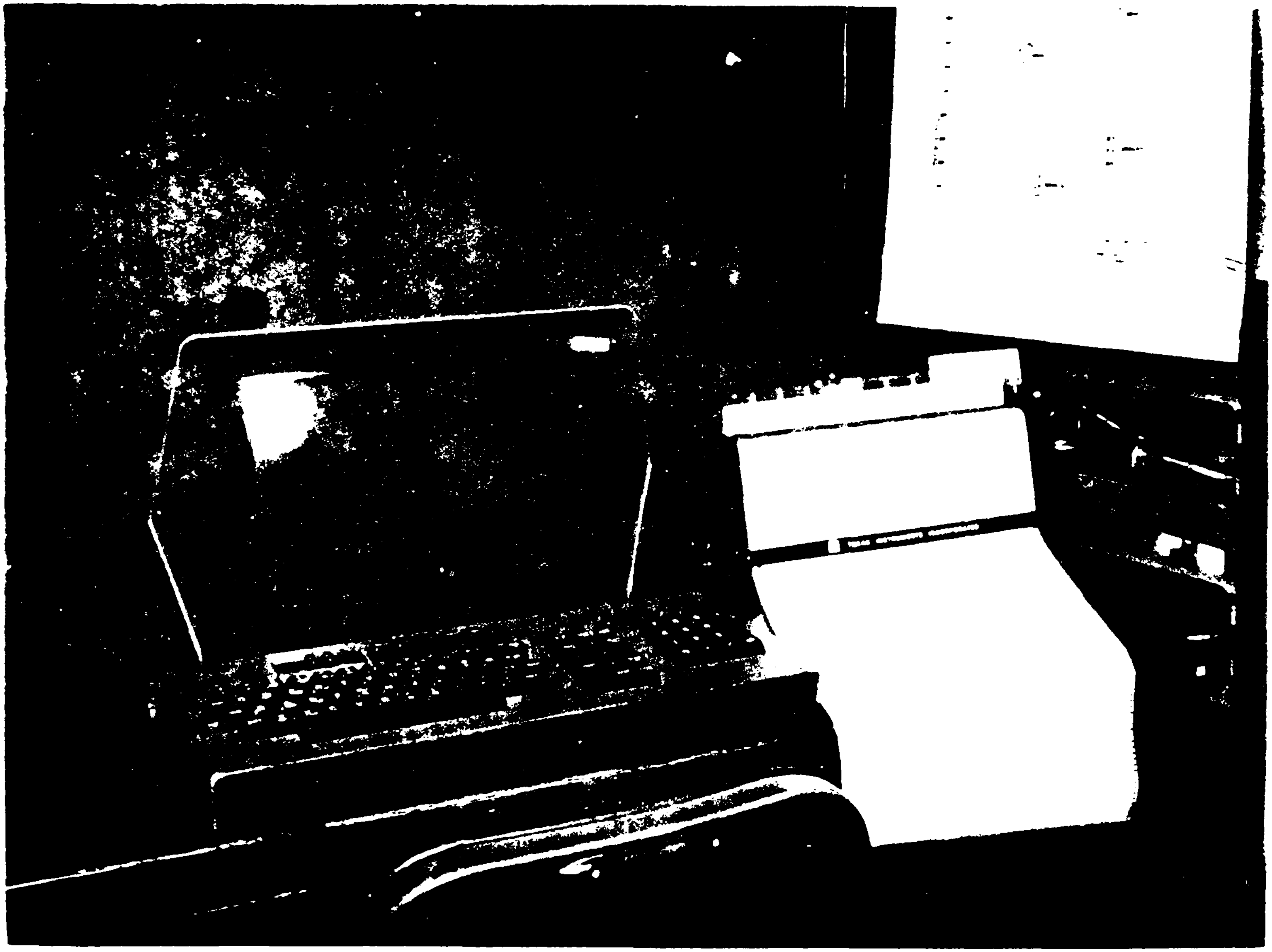


FIGURE 10. THERMAL RECORDER AND CRT

FIGURE 11. SMS SIMULATION FACILITY

operators selected the method prescribed by the Sensor Intelligence School in arriving at their results. Calculations made by the operators were based on the following equation.

$$LC = \frac{D}{M} \times TTI - CDR$$

where: LC = Length of column

D = Distance between two sensors

TM = Meantime between the two sensor patterns

TTI = Total time of the first sensor pattern

CDR = Combined detection radius of the two sensors.

Distances between vehicles was assumed (and actually was) to be 100 meters. Operators would calculate the number of objects in a column by the equation:

$$\text{Number of objects} = \frac{\text{Length of Column}}{\text{Assumed distance between objects}}$$

The operators obtained direction of the objects from observing which sensors activated in a string and from using the sensor deployment map (Figure 6).

Test Tactical Situations

The tactical situations used for operator testing were composed mainly of columns of track and wheel objects. The actual columns and their object mixes are shown in Figures 4 and 7. Here, the object number, type, string, velocity, and start time are defined for each object. Also, groups (columns) of objects are obtained by placing individual objects on the same string separated by a fixed distance. The target tracks and sensor field are the same as in Figures 2 and 6.

Several features of the objects used for the tactical situations should be noted. First, most objects used in the test were travelling at 600 meters/minute and separated by 100 meters in columns. The speed variations from these figures are noted in the results. Secondly, both mixed and unmixed columns were used; that is, columns with either more than one type object or columns with only one type. Third, mixtures were made of varying proportions of track or wheel targets in individual columns.

In general, the tactical situations involved a large number (35 - 40) of objects converging and spearheading towards the forward edge of the battle area. As can be seen in Figure 2, which shows the trail layout, the objects had many paths on which to travel and be detected by the various strings of sensors.

Test Data Definitions

The data from each individual operator trial was tabulated and further reduced to several measures. The following is a list and definitions of each test data item displayed in Tables 1 and 2:

1. Percentage correct object count: The overall accuracy of the calculated total number of objects in a tactical situation is described by a parameter. It is defined as:

$$1.0 - \frac{\sum |\text{count error}|}{\sum \text{number of all objects}}$$

The numerator of the second term is the sum of the absolute differences between the actual number of objects in a tactical situation and the calculated count. The denominator is the sum of all actual target objects.

2. Percentage ROS accuracy: The calculated rate of speed (ROS) in meters per minute was measured for accuracy by the following, for each operator:

$$\frac{\overline{\text{calculated ROS}}}{\text{actual ROS}}$$

Note that the actual ROS is constant for each data grouping and hence the accuracy is simply the average of the calculated rate of speed divided by the constant actual ROS.

3. Percentage direction accuracy: The directs of the columns as noted by the operators were compared to the actual directions and measured for each trial by the following paramenters:

$$\frac{\text{number correct directions}}{\text{total of direction attempts}}$$

The numerator is the total number of correct directions chosen by the operator while the denominator is the total number of attempts by the operator to determine object(s) direction. Note that not all changes in the target's direction were observed and hence were not included in this percentage.

4. Percentage correct mixed target determination: This quantity measures the performance of an operator to correctly separate a mixed column of objects into the correct object mix:

$$\frac{\text{number of correct mixed target determinations}}{\text{total mixed target determination opportunities}}$$

INDIVIDUAL OPERATOR (TRIAL)	CORRECT OBJECT COUNT %	ROS ACCURACY %	DIRECTION ACCURACY %	CORRECT MIXED TARGET DETER MINED %	ALARM CLUSTERS PROCESSED %	CORRECT TRACK OBJECT COUNT %	CORRECT WHEEL OBJECT COUNT %	CORRECT TRACK OBJECT COUNT (MIXED) %	CORRECT WHEEL OBJECT COUNT (MIXED) %	ACTUAL OBJECT ROS MTRS/MIN	SENSOR DET ECTION RADIUS METERS	DISPLAY FORMAT	OBJECT TAPE USED
A	40.0	89.3	50.0	N/A	25.0	N/A	40.0	N/A	N/A	420	500	1	42
B	40.0	56.4	N/R	N/A	33.3	N/A	40.0	N/A	N/A	420	500	2	42
C	80.0	64.3	N/R	N/A	33.3	N/A	80.0	N/A	N/A	420	500	1	42
D	0.0	71.9	N/R	N/A	33.3	N/A	0.0	N/A	N/A	420	500	2	42
A	66.7	48.7	100.0	N/A	25.0	66.7	N/A	N/A	N/A	720	500	1	42
B	33.3	20.3	N/R	N/A	25.0	33.3	N/A	N/A	N/A	720	500	2	42
C	100.0	24.3	0.0	N/A	25.0	100.0	N/A	N/A	N/A	720	500	1	42
D	0.0	23.1	N/R	N/A	25.0	0.0	N/A	N/A	N/A	720	500	2	42
A	79.2	51.9	100.0	0.0	19.4	89.0	N/R	-26.0	N/R	600	500	1	42
B	39.2	49.9	100.0	0.0	30.5	78.0	N/R	25.0	N/R	600	500	2	42
C	58.3	47.4	100.0	0.0	33.3	66.7	N/R	39.0	N/R	600	500	1	42
D	0.0	43.9	40.0	0.0	24.3	0.0	N/R	0.0	N/R	600	500	2	42

N/A Not applicable
N/R No response

TABLE 1. OPERATOR RESULTS – GROUP 1

INDIVIDUAL OPERATOR (TRIAL)	CORRECT OBJECT COUNT %	ROS ACCURACY %	DIRECTION ACCURACY %	CORRECT MIXED TARGET DETERMINED %	ALARM CLUSTERS PROCESSED %	CORRECT TRACK OBJECT COUNT %	CORRECT WHEEL OBJECT COUNT %	CORRECT TRACK OBJECT COUNT (MIXED) %	CORRECT WHEEL OBJECT COUNT (MIXED) %	ACTUAL OBJECT ROS MTRS/MIN	SENSOR DETECTION RADIUS METERS	DISPLAY FORMAT	OBJECT TAPE USED
E	50.0	86.9	100.0	N/A	66.7	N/R	N/A	N/A	N/A	420	200	1	42
F	66.7	83.3	100.0	N/A	50.0	66.7	N/A	N/A	N/A	420	200	1	42
G	66.7	83.3	100.0	N/A	50.0	33.3	N/A	N/A	N/A	420	200	1	42
E	76.0	95.8	100.0	N/A	67.5	N/A	N/R	N/A	N/A	720	200	1	42
F	N/R	97.2	100.0	N/A	33.3	N/A	N/R	N/A	N/A	720	200	1	42
G	60.0	53.2	100.0	N/A	33.3	N/A	60.0	N/A	N/A	720	200	1	
E	54.0	88.7	94.4	100.0	56.3	N/R	N/R	33.3	33.3	600	200	1	42
F	48.3	81.7	100.0	83.0	41.7	75.0	N/R	27.0	39.7	600	200	1	42
G	70.3	74.2	90.9	60.0	40.7	44.3	N/R	67.4	38.2	600	200	1	42
L(G)	58.6	84.1	90.9	55.6	37.9	50.0	N/R	41.4	40.8	600	200	5	49
M	54.7	87.5	100.0	80.0	35.3	75.0	N/R	17.0	48.4	600	200	3	49
N(G)	50.0	82.9	100.0	50.0	50.0	50.0	N/R	-15.5	20.0	600	200	4	49

N/A Not applicable
N/R No result

TABLE 2. OPERATOR RESULTS – GROUP 2

The numerator is the total number of times a single operator correctly separated a mixed column into its individual object types while the denominator indicates the total number of opportunities an operator had to determine mixed targets which he processed when they were presented.

5. Percentage alarm clusters processed: For all cases during the tactical situations, a number of adjacent alarm clusters were generated by the objects while passing the sensor strings. Many times a direction change occured at one portion or another in the object track. The variable which is measured here attempts to give an indication of the usage of available data which was presented to each operator. It is defined as:

$$\frac{X + 1}{Y - 1}$$

where X = number of overall operator calculations made on the target alarm clusters produced by a sensor string.

Y = total number of alarm clusters produced by a target in a sensor string.

6. Percentage correct object type count: A measure of the count accuracies was made on both single and mixed object columns. It was calculated for each object type classification attempt made by an operator:

$$1.0 - \frac{\sum \text{percent count error}}{\text{total count attempt}}$$

The numerator is the percent count error for each count attempt made on a single object type. The denominator is the total number of count attempts made on the same object type. For example, suppose an operator calculated there were three track and one wheel targets when, in reality, there were four track and four wheel targets. His percentage correct object type count for track and wheel would be 75% and 25%.

Note that this parameter is calculated separately for mixed and single object columns in the data. A separate data column for both single and mixed target types is given.

7. Actual Target ROS: The actual rate of speed (ROS) of the targets is given in meters per minute. Note that all targets (columns) are composed of multiple objects.

8. Sensor detection radius: The sensor detection radius for a track vehicle is given in meters.

9. Character/Symbol displayed: The type of classification displayed characters/symbols during the testing is shown in Figure 12. Each format which was used for an individual operator trial is given by the corresponding set number in the data.

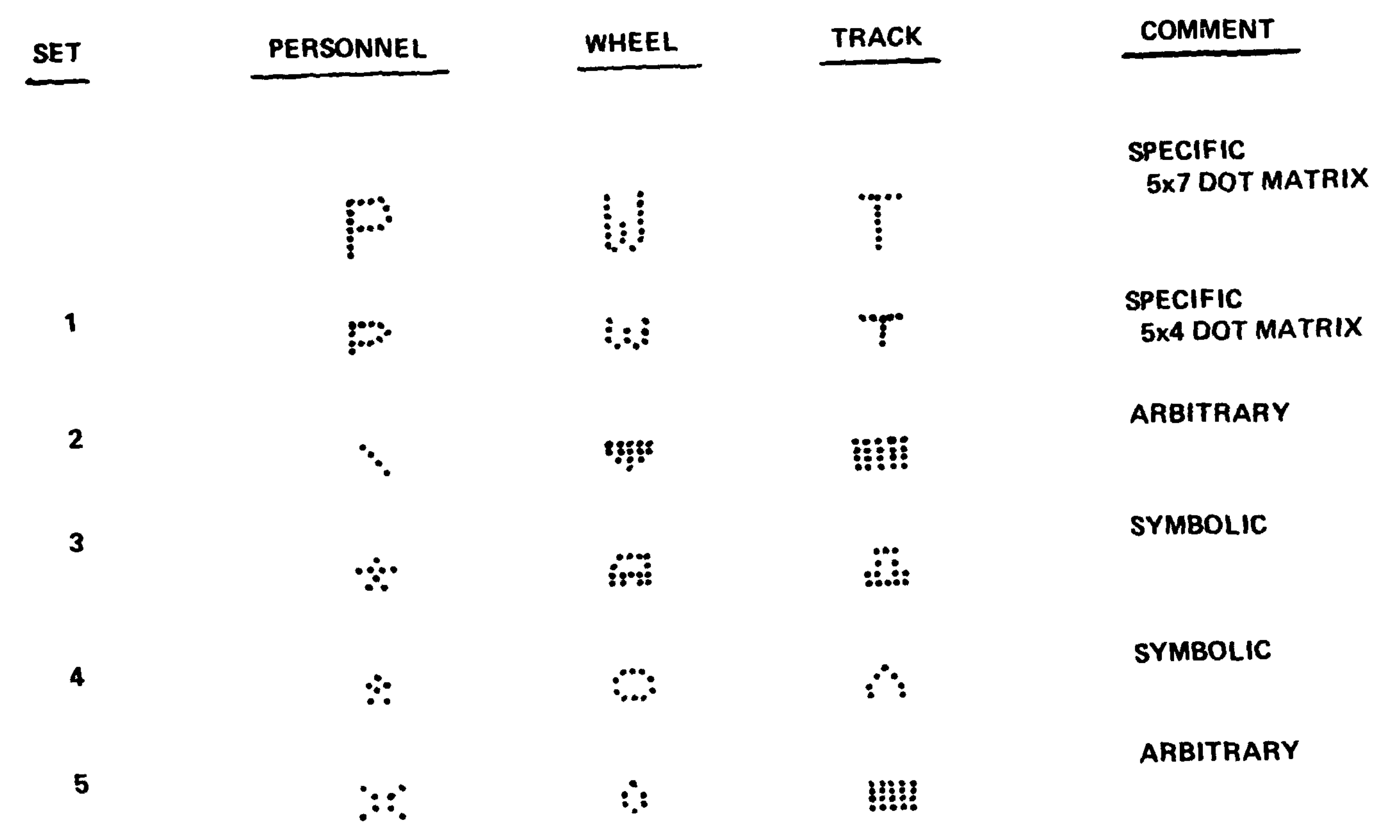


FIGURE 12. CHARACTER/SYMBOL REPRESENTATION

10. Object used: There were two sets of objects used during the testing. They are shown in Figures 4 and 7. The individual targets are partitioned in the data by their ROS. Note that the majority of all targets is moving at 600 meters/minute.

OPERATOR RESULTS

The operator results are presented in Tables 1 and 2. They consist of results from two operator groups: one and two. Group one operators had a 500-meter sensor detection radius whereas group two had a 200-meter radius. There are also differences among both groups in the target ROS. This data, although small, is separated from the majority (600/minute).

Several features can be noted from the presented data: (1) all the operators of group one had difficulty in calculating time on the thermal printout and hence there are errors in their ROS calculations. They used twice the actual value of time, although they received detailed instructions on the procedure to calculate time; (2) the large detection radius used by group one results in 0% correct mixed target determination in all cases, whereas the lower detection radius of group two produces over 50% correct mixed target determination in all cases; (3) all the operators were consistent in their calculations of correct object count, direction accuracy, and ROS (if (1) above is taken into account); (4) the operators consistently did not use all the recorder data available to them as shown in the percentage of alarm-clusters-processed column.

Operator's Comments

This section discusses useful comments made by the operators and the observations of the operators by testing personnel. They relate mainly to the printing of the classification symbols from the recorder:

1. As long as the character types or symbols were (a) completely distinguishable from each other, and (b) had no overlap, there was no classification confusion.

2. Characters or symbols should be distinguishable not only as in 1, above, but also when embedded in groups of other characters or symbols.

3. With an 80% correct classification response, the operators had no problem in choosing the correct majority classification. This resulted in a saving of analysis time on the part of the operators, since they did not need to estimate the type object, and it eliminated the corresponding problem of choosing the type object in a column of mixed objects by velocity discrimination.

4. Shortened or squatted 5 x 4 character formats were legible.

5. Common characters or symbols which cause automatic associations with object types are the best. Otherwise, the operators had to learn and constantly refer to the symbol drawings for their meaning. Once they were thoroughly learned, however, there seemed to be no difference among the characters or symbols.

Operator Problem Areas

Several problem areas were exposed during the testing of the operators. They are mentioned in summary form to indicate possible areas in which improvements can be made in operator performance:

1. Different sensor detection radii, other than the assumed values which the operators use, cause large errors in their calculations.

2. Large inconsistencies in the results of several operators were found. For example, when one operator, M, calculated the object count for the same target, he obtained values of 2, 8 and 11 objects. Note that this is from alarm clusters which are identical in length, with identical sensor characteristics.

3. Targets were not tracked. There was no observed written grouping together of identical targets on the printout. Thus, there was no accumulation of knowledge about the individual targets and hence updates of their characteristics, as would have been helpful in 2 above, during their passage through the monitored area.

4. Many changes in target direction were not observed by the operators. For example, if there were several alarm clusters and the last two indicated a turn in the target direction, most operators ignored this fact. They only processed, as a rule, the first two alarm clusters.

5. Operators' performance was, in general, inflexible; they could not compensate for changing conditions which could be encountered in real tactical situations.

Recorder Observations and Recommendations

The thermal recorder print heads consisted of 80 individual fixed heads made up of five in-line dots each. The use of individual heads created problems in maintaining equal contact pressure between the heads and the chart paper. This caused non-uniformity in the shading of the characters so that some characters would be light and difficult to read while others would be dark and easy to read. Though this created no problems during the test because pens which printed dark characters were selected, it may be an inherent problem associated with fixed head printers and should be considered before selecting a recorder for the SMS.

The recorder required the use of a non-standard size chart paper. The paper Texas Instruments used during the development of the recorder was not inscribed with columns or rows; however, this paper was used because a small number of rolls were required and the cost to manufacture paper of the correct size and markings was too high. Separating the groups of pens representing the individual sensor strings and marking the pen positions in the recorder near the print heads enabled the operators to match the activation to the corresponding sensor.

One problem was calculating time because there were no row markings to act as a reference. The operators were given a formula which required measuring distances and multiplying by a conversion factor. This method

worked well with one group of operators; however, the other group consistently calculated speeds of approximately one half the actual target speed.

It is recommended that the SMS chart paper have some form of timing reference such as a row indicator, a timing tick mark, or the actual time printed by the processor.

APPENDIX A

SYSTEM OVERVIEW

Software System Description —— The software system is structured using a real time operating system (RTOS). Some of the more important features which this allows are:

a. Multiple programs operating concurrently with interleaving.

b. Program priorities determine execution sequence and distribution of processing time.

c. Accurate time of day (clock) available.

d. Programs can communicate and pass data among themselves.

e. Programs can be segmented and overlayed from disc.

A simplified diagram of the system is given in Figure 13. The center of the system is the executive. All system interrupts, device interrupts, and system service requests are handled here. The executive always goes to the scheduler once it is done. The scheduler determines which user program: Task 1, Task 2, etc., is to be activated. This depends on both the current state of the tasks as well as their priority. Once the scheduler starts a program it continues in operation until the next interrupt to the executive occurs.

The drivers indicated at the bottom of the figure are software routines which interface the device controllers to the executive. The system uses them for all I/O operations performed through their respective devices.

System Hardware Description —— The Sensor Monitor Set (SMS) simulation used an Interdata Model 70 minicomputer as the processor with standard peripheral equipment and a Texas Instrument Thermal Recorder. The Hardware-System Configuration (Figure 14) depicts all the hardware used in the SMS Test. Note, the thermal recorder is not a standard peripheral item and required the development of special interface circuitry.

The processor controls all activities and performs all arithmetic and logical functions. It executes instructions in an ordered sequence to complete a specified task or program. The processor has 16 hardware registers for data manipulation, hardware divide/multiply and floating point instructions. The main memory is core memory with a capacity of 64 k bytes of which 48 k bytes are presently installed. The selector channel is a standard direct memory access device that allows connection of high speed peripheral devices directly to main memory. The maximum data transfer rate is 2 mega bytes per second. All medium to low speed devices are connected to the Multiplexer Input/Output bus. This is a request/response bus consisting of 30 lines: 16 bi-directional data lines, 8 control lines, 5 test lines, and a system initialize line.

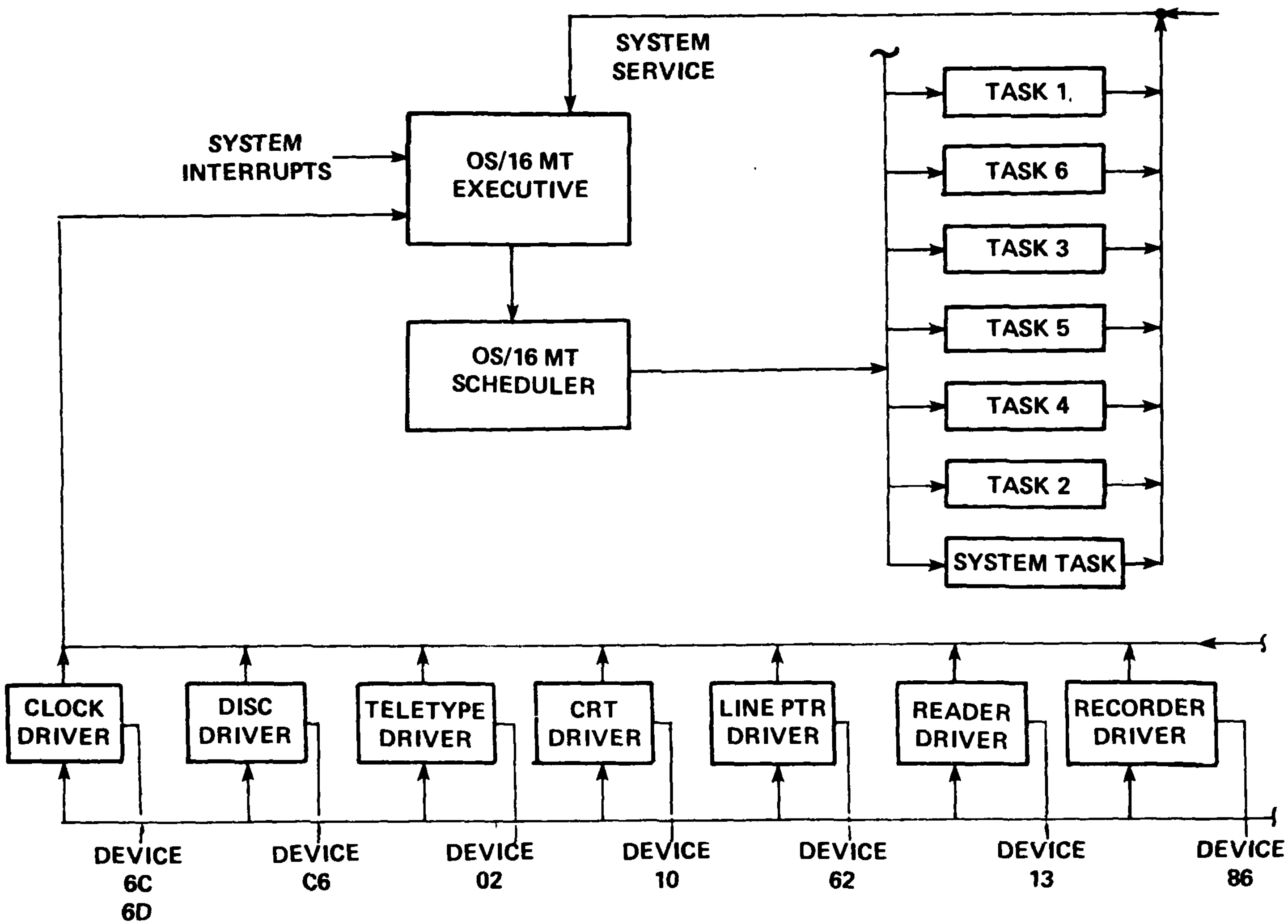


FIGURE 13. RTOS SOFTWARE SYSTEM DIAGRAM

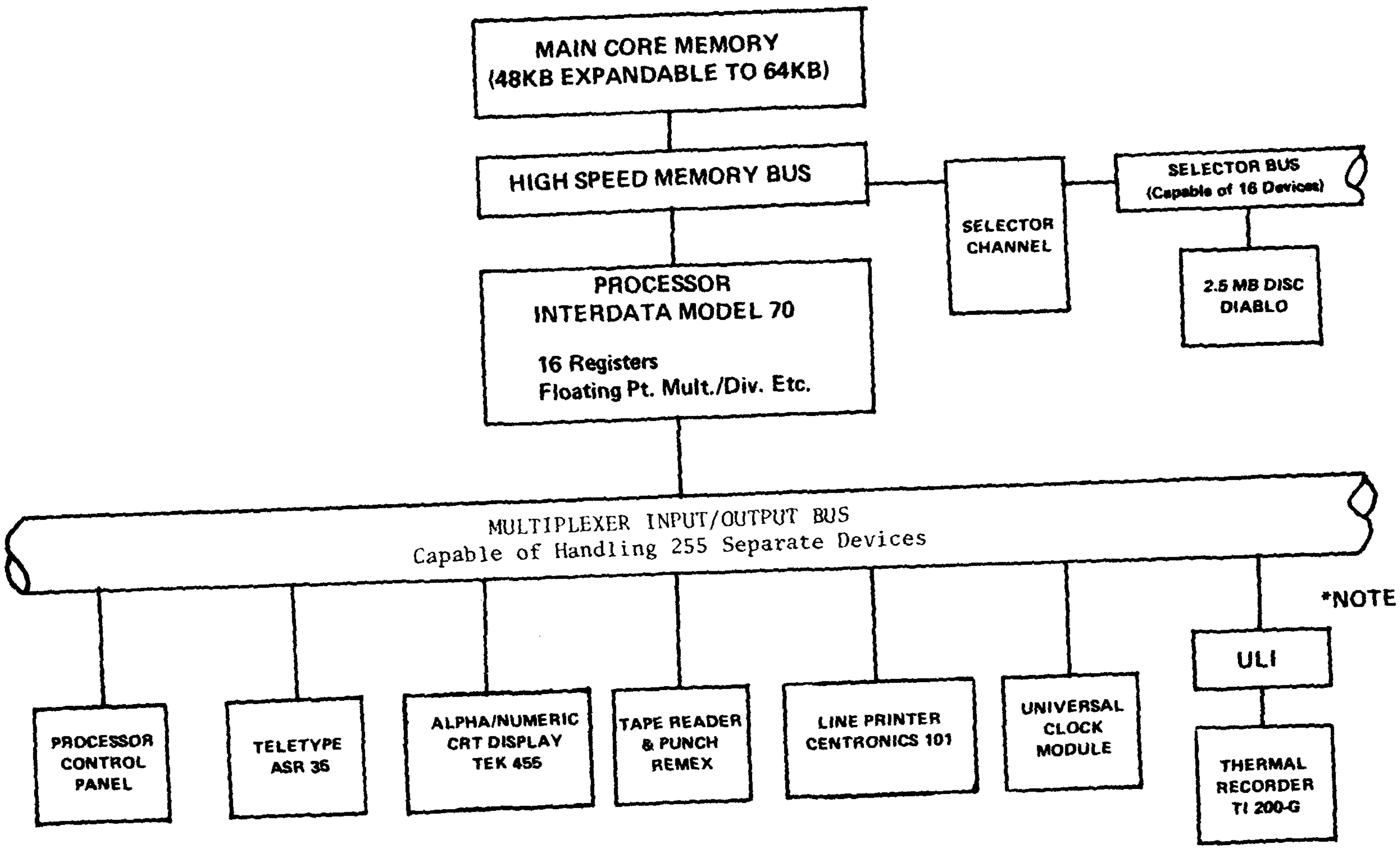


FIGURE 14. HARDWARE SYSTEM CONFIGURATION

Interrupt detection and hardware vectoring can be accomplished for all of a possible 255 devices which can be interfaced to the multiplexer bus. The peripherals used are standard devices offered through Interdata with the exception of the thermal recorder (TI Graphic 200) which was obtained through Texas Instruments. The recorder was integrated into the system using a Universal Logic Interface Board and Associated Hardware to obtain the specified input and output characteristics required by the multiplexer bus.

APPENDIX B

DETAILED SOFTWARE DESCRIPTION

Operating System

The real time operating system used is OS/16 - MT (multi-tasking operational system). It is divided into system programs and user program units called tasks. System programs include the executive, scheduler, initialization route, I/O drivers, and interrupt handlers.

Tasks

A task can be a single program or a group of programs, whose execution is controlled by the operating system. Each task exists in one of eight states; they are:

(1) Dormant - The task has not been started or has gone to completion.

(2) Active - The task which is currently executing instructions. Only one task can be in this state at any given time.

(3) Ready - The task which will start or resume execution when it becomes the highest priority ready task.

(4) Task Wait - A task has called another task into execution and is waiting for the called task to go to completion.

(5) Console Wait - The task is waiting for an operator reply.

(6) I/O - The task is waiting for a specific time interval to elapse.

(8) Overlay Wait - The task is waiting for an overlay to be loaded.

Each task is assigned a priority level based upon which task has operating privileges over the other tasks. The priority numbers are selected with the highest priority task, always being the command processor task, having the lowest number, and the lowest priority user task having the highest number (task to be run last). The following are the Sensor Monitor Set Systems tasks in order of their priority of execution:

Priority	Name of Talk
0	Command Processor (within OS-16-MT)
1	Thermal Recorder Task (Task 1)
2	Tactical Situation Simulator Task (Task 6)
3	Input Processing Task (Task 3)
4	Operator Processing Task (Task 4)
5	Command Directory Task (Task 5)
6	Line Printer Task (Task 2)

Executive

The OS/16-MT executive is a collection of routines that are entered as a result of internal interrupts. These interrupts include supervisor calls, illegal instructions, arithmetic faults, I/O termination, I/O queue overflow, and console interrupts. The executive always exits through the task scheduler. Normally, the status of a least one task is changed by the executive in servicing the interrupt. This means that the task that was active at the time of the interrupt may no longer be the highest priority ready task when the executive exits. When it exits, the scheduler decides which task is to be activated.

Real Time Clock (Universal Clock Module)

The OS/16-MT system maintains two clocks, a time of day clock and an interval timer. The time of day counter is a full word count kept in seconds since midnight. It is driven by a presettable 120 Hz interrupt from the Universal Clock Module. This counter is initialized to zero on system start up and may be set through the operator command to set time. From this counter, a task may request the current time of day or that it be placed in a time wait until a specified time of day is reached. A task may also request that it be placed in a time wait for a specified time interval.

TASK 1

The recorder input buffer is eight 5-bit characters long and must be filled with either activation data or zeros when there are no activations. When the recorder is ready to receive new data, it sends a recorder Not Busy signal to the computer. This signal interrupts the processor. The interrupt routine conditions the state of the operating system so that the scheduler starts Task 1.

Task 1 controls the interface between the computer and the recorder. It designates the input buffers to be filled with new activations data, the buffer from which data is to be put out and it controls the actual output of the data.

Thermal Recorder. The thermal recorder is capable of printing an 80 character line. The print head is a row of eight 5-bit stationary thermal heads with a space between each group of 5 bits. A 5 by 7 dot matrix character is printed by building the characters one row at a time. The applicable dots of the bottom are printed by driving the corresponding bits of the print head. The chart is moved slightly, and the applicable bits of the same 5 bits are driven to print the second row and so on until all seven rows have been printed.

The row of print heads is divided into four sets of twenty 5-bit subsets each. This permits using a smaller power supply for the printing drivers. Each set is driven at different times. One set is driven, and the chart is moved a small step in order to move the burned portion from under the head, then another set is driven and another step taken. This procedure continues until all four sets have been driven and four steps have been taken. At this time one row has been printed.

The recorder logic contains a character generator which requires as an input the 8-bit ASCII code for the desired character; however, this was inadequate for the recorder's intended use. Other characters or symbols in addition to those available in the character generator were required. It was also required that on-the-spot character and symbol configurations be changed so that any configuration capable of being generated by a 5 by 7 dot matrix would be available.

Modifications were made in the recorder and an option board built which, in essence, removed the character generator from the recorder. This function was designed into the software in the Model 70. The computer outputs to the recorder one row of a character at a time. Changes in software requiring only a few minutes can enable the generation of symbols in any 5 by N dot matrix.

Chart speed is controlled by the recorder; however, switches enable various speeds to be set into the recorder. Additional speeds can be acquired by using an external oscillator. The switches and the external oscillator enables speeds being considered for the SMS to be obtained.

TASK 2 - The Line Printer Task

The function of the line printer task is to output the sensor activations to the line printer correctly formatted with a heading printed approximately every thirty sensor activations (Figure 15). The following data is recorded for each activation:

RID - Receiver and Sensor Identification Number

TYPE - Sensor Type: Examples

SFE - Seismic Feature Extractor

VFP - Variance Frequency Processor

RID	TYPE	EAST	NORTH	TIME	TBR
101	SFE	004389	028832	000217	TRK
101	SFE	004389	028832	000227	TRK
101	SFE	004389	028832	000237	TRK
102	SFE	004500	028500	000240	WHL
101	SFE	004389	028832	000247	TRK
102	SFE	004500	028500	000250	TRK
101	SFE	004389	028832	000302	TRK
102	SFE	004500	028500	000302	WHL
103	SFE	004657	028132	000306	TRK
101	SFE	004389	028832	000312	WHL
102	SFE	004500	028500	000312	TRK
103	SFE	004657	028132	000316	TRK
101	SFE	004389	028832	000322	TRK
102	SFE	004500	028500	000322	TRK
103	SFE	004657	028132	000326	TRK
105	SFE	005066	028201	000326	WHL
102	SFE	004500	028500	000332	TRK
113	SFE	013000	017050	000332	WHL
103	SFE	004657	028132	000336	TRK
105	SFE	005066	028201	000336	TRK
102	SFE	004500	028500	000342	TRK
113	SFE	013000	017050	000342	TRK
103	SFE	004657	028132	000346	TRK
104	SFE	004094	027581	000346	TRK
105	SFE	005066	028201	000346	TRK
113	SFE	013000	017050	000352	TRK
103	SFE	004657	028132	000356	TRK
104	SFE	004094	027581	000356	WHL
105	SFE	005066	028201	000356	TRK
629	SFE	013000	017500	000402	TRK
113	SFE	013000	017050	000402	TRK
RID	TYPE	EAST	NORTH	TIME	TBR
103	SFE	004657	028132	000406	TRK
104	SFE	004094	027581	000406	TRK
629	SFE	013000	017500	000412	TRK
113	SFE	013000	017050	000412	TRK

FIGURE 15. LINE PRINTER ACTIVATION OUTPUTS

EAST - East UTM Coordinates

NORTH - North UTM Coordinates

TIME - System Time the Activation was processed

TBC - Sensor Data Processed

Class I Detection Only - (Blank) - No Data

Class II Classification - TRK - Track
WHL - Wheel
MAN - Personnel
UNK - Unknown

The line printer works in conjunction with the line printer driver, which outputs each individual character to the line printer.

TASK 3 - Input Processing Task

The function of the Input Processing Task is to process the activations and associated data from the tactical situation simulator task (Task 6). When activation data is passed to Task 3, the following programmed sequence occurs:

1. Validation and checking of sensor identification codes and sensor data for a particular class of sensor

Type I - Detect Only Sensors - No Data

Type II - Classification Sensors - Classification Data.

If the data obtained from a classification sensor is not within proper parameters, the activation is tagged with a corresponding symbol for bad data.

2. Decoding and proper formatting of the activation and associated data for transfer to the peripheral equipment used by the computer.

3. Setting up of the data into specific buffer location to be used by Task 1 to output this data to the Texas Instruments Thermal Recorder. This data can be displayed in any format by the character generator routine.

4. Outputting of the activation to the following devices:

a. Disc, on which a history record is kept for all activations.

b. Cathode Ray Tube, for display in simulated map format.

Upon completion of these functions, the task terminates itself and becomes dormant until another activation is passed to it.

Input Data Processing Routine for Texas Instruments Thermal Recorder

The data processing routine for the Texas Instruments thermal recorder tests and validates sensor input data received through the Input Processing Task (Task 3). This routine formats and stores data to be outputted to the recorder by the Thermal Recorder Task (Task 1). The sensor activations are generated by the Tactical Situation Simulator Task (Task 6) which simulated sensor activations, sensor data and sensor time of activations. These simulated activations and associated data are then stored into the Data Processing Queue Facility (Input Queue) which is a circular list processing storage area (first-in-first-out) located within Task 3. When sensor activations and data enter the queue, the input processing task and data processing routine for the thermal recorder begin execution of their programs. When the data processing routine for the thermal recorder is executed (Figure 16), the following programmed functions occur:

1. Determine the corresponding sensor type from the activation's sensor identification and channel number (RID#). Two types of sensors generated activations:

   a. Type I - Detection-Only Sensors

      Channel No. + Sensor ID No. + Time of Activation

   b. Type II - Classification Sensors

      Channel No. + Sensor ID No. + Time of Activation + Classification Data

2. Decode and store the data into an output storage area which will be accessed by the thermal recorder task. This data can be translated into the form of single data bytes (detection-only activations) or full characters or symbols (classification activations).

The routine begins by determining the type of sensor activation which is being processed. When the data enters, it consists of a channel and sensor identification number (RID). The RID is checked against a list of known active sensors with RID numbers stored in a common data base located in the core memory. Each RID number in the data base has a corresponding sensor type, either detection-only (Type I) or classification (Type II). When the sensor type has been determined, the appropriate data handling routine is executed under program control.

Detection-Only Activation - Type I Sensor Routine

Data from a detection-only type sensor is passed to the Type I Sensor Routine. This routine loads the last time of activation for that particular sensor identification number (RID#). The last time of activation of each of the active sensors is stored in the common data base of the input processing routine. The routine loads the current time of activation from the input queue where the current activation is stored temporarily until servicing of the activation is completed. A time comparison routine is used to determine if the difference between the last activation time and the present activation time for the specific RID number is less than, or greater than, a fixed differential in time which can be specified. The time differential used in the SMS software was 1 minute between successive activations. This was done so that activations

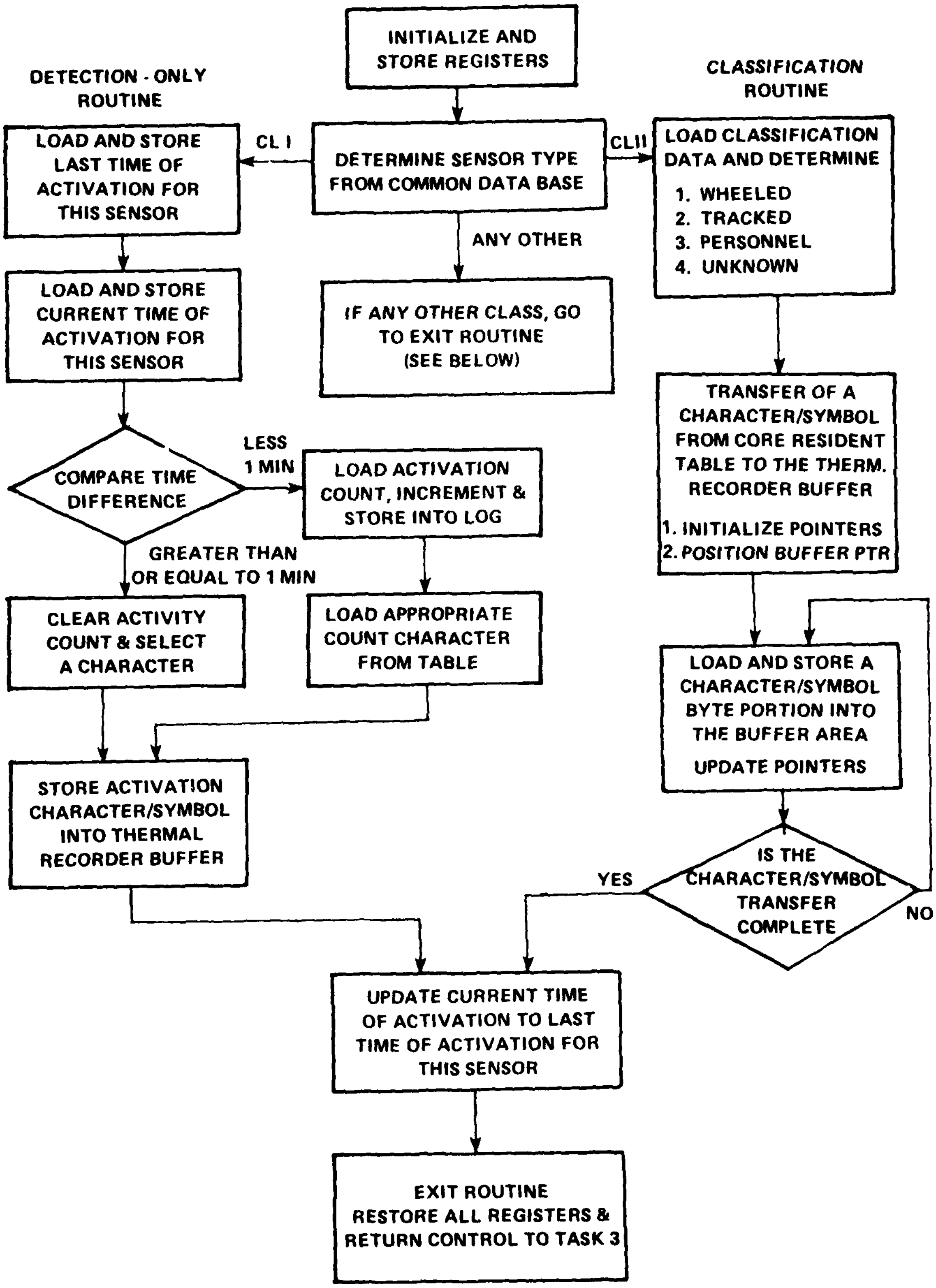


FIG. 16 THERMAL RECORDER DATA PROCESSING FLOW DIAGRAM

arriving within this time limit, successively, would be given different symbol representations to be displayed to the recorder, thereby isolating single activations and multiple activations at the recorder output so that possible false activations could be recognized at a glance (see Figure 17). The technique for displaying the data follows:

a. If the differential in activation times is less than one minute, the activation log counter for the sensor identification number is loaded, updated and restored in the activation log. A symbol is taken from the character/symbol table corresponding to the updated activation count and stored in the appropriate recorder pen number location in the recorder output buffer.

b. If the differential is greater or equal to one minute, the activation log count for this RID number is initialized to Zero, stored into the activation log, and a symbol is taken from the character/symbol table corresponding to an initial activation. This symbol is a single dot, which is stored into the recorder pen number location, in the recorder output buffer for this RID number.

When this has been completed, the current time of activation is stored where the last time of activation was in the common data base, thereby making the current time the last time of activation for this sensor identification number. Following this, all the registers of the operating system are restored and the processing of the input data is continued on the input processing task, where software control is returned.

Classification Sensor - Type II Sensor Routine

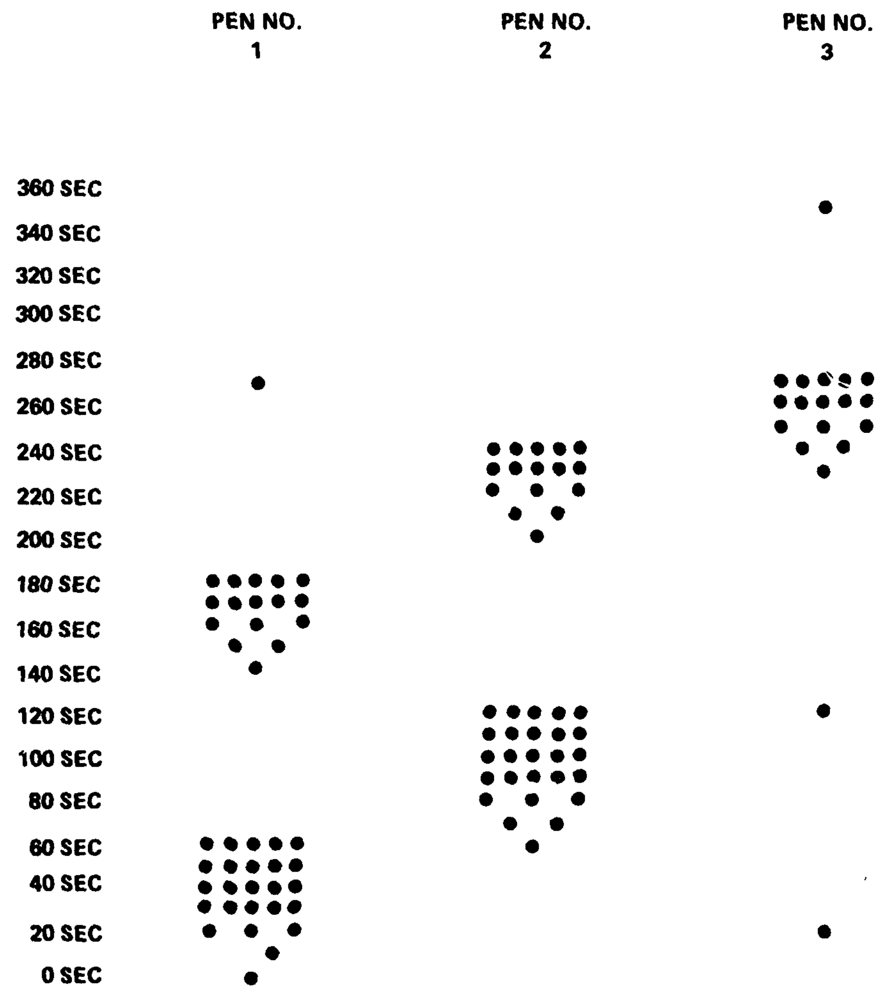
When a classification sensor activation is determined, the data portion of the activation is loaded from the input queue where the current activation is temporarily stored. The data is decoded and checked for its classification. In the simulation tests, three different types of valid classification data were generated and processed through the operating system. The valid classifications were:

a. Tracked Vehicle

b. Wheeled Vehicle

c. Personnel

If data received for an activation does not coincide with any of the valid data codes, the activation is tagged with a symbol signifying that the data is unknown (i.e., insufficient data to correctly classify the target). After the activation and decoded data have been verified for type of classification, the program goes to a reference table in the core memory and determines which character/symbol is to be stored into the recorder buffer to be displayed. For example, if the data for an activation was found to be that of a tracked vehicle, the routine would go to the reference table and select a "T" for tracked vehicle. Then the character/symbol would be transferred into the recorder output buf f



NOTE: Symbol initialization after
60 sec. of no activity

FIG. 17 SYMBOL GENERATION FOR DETECTION-ONLY SENSORS

specified by the recorder pen number for this particular sensor identification code. Similarly, if the activation data was that of a wheeled vehicle, a "W" would be selected from the table. Having the character generation under software control expands the possibility of outputting any 5 by 7 character or symbol rather than the standard ASCII Code, or, for that matter, any 5 x N character or symbol. (See Figure 18.) This technique is used to shrink the characters to a 5 x 4 dot matrix thereby enabling slower recorder chart speeds.

When the character/symbol has been completely transferred to the recorder output buffer, the current time of activation is stored into the common data base for this sensor identification number; all registers are restored to initial entry values, and program control is transferred back to the main input processing routine.

Recorder Output Buffer Data Structure

The data base structure for the Texas Instruments thermal recorder is a core resident storage area made up of seven 80 8-bit data strings. (See Figure 19.) The first three most significant bits of each data byte (8 bits) are not required by the recorder and are only used to simplify the software data base and programming of the data transfer.

The seven data strings, or data blocks, represent the seven character/symbol lines as discussed in the character/symbol generation of classification sensors (Figure 18). The seven data blocks are configured in a circular list, data is transferred sequentially into all seven data blocks, continuing into the first block after the last has been filled. The starting data block is determined by the relationship between the recorder output state and the time an activation is being processed.

The data block structure uses seven data blocks to generate 5 by 7 dot matrix characters/symbols. By extending or shortening the circular data structure, any 5 by N character/symbol can be generated. This technique was used in changing the size of the characters for slower chart speeds used by the thermal recorder.

TASK 4 - Operator Command Processing Task

The purpose of the operator command processing task is to enable the operator to execute disc overlay programs. The task consists of a four kilobyte overlay area (storage area) and some controlling software. There are many programs at the operator's disposal (Edit, Admin, etc.), and storing all of them into the core is impossible owing to restraints on the core size. Also, additions and modifications to software available to the operator would require a regeneration of the entire system. Therefore, all of the interactive programs were stored on a disk file non-resident to the system. When a program is requested by the operator, the non-resident disc file is searched (Task 5), the program is loaded into the overlay area (Task 4), and program control is passed to the non-resident routine which is in the overlay area by starting Task 4. Upon completion of the non-resident routine 4 terminates itself and becomes dormant.

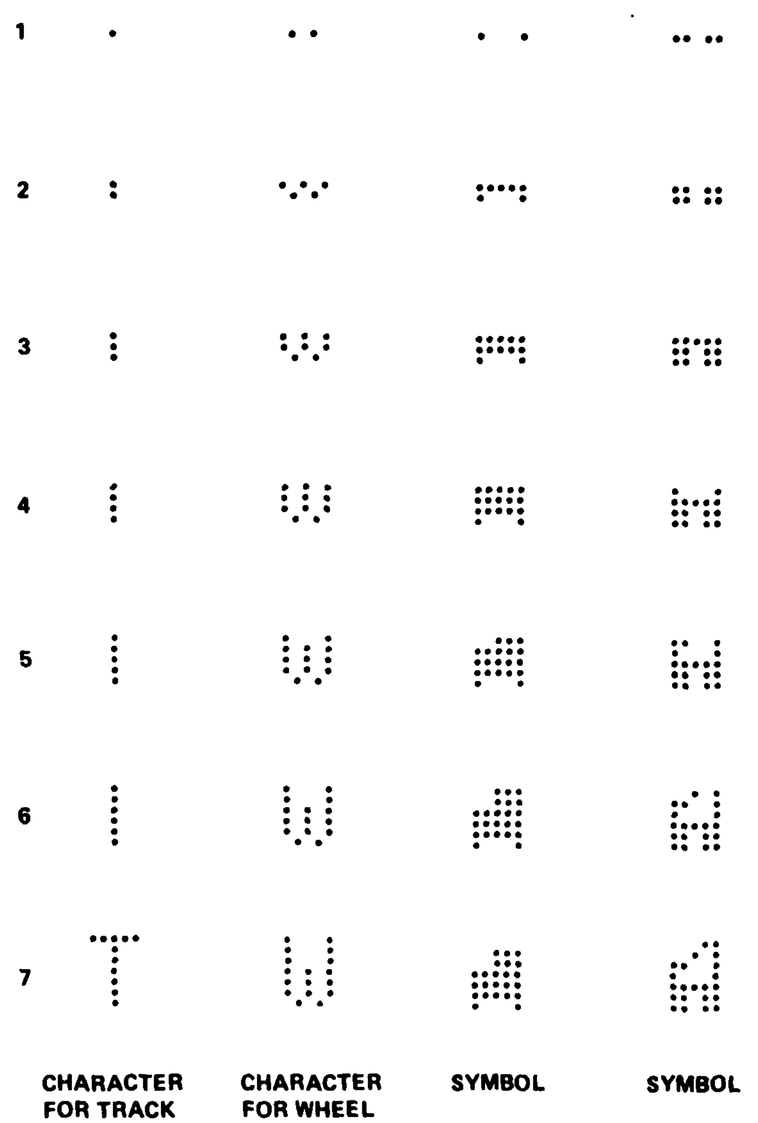


FIG. 18 CHARACTER/SYMBOL GENERATION FOR CLASSIFICATION SENSORS

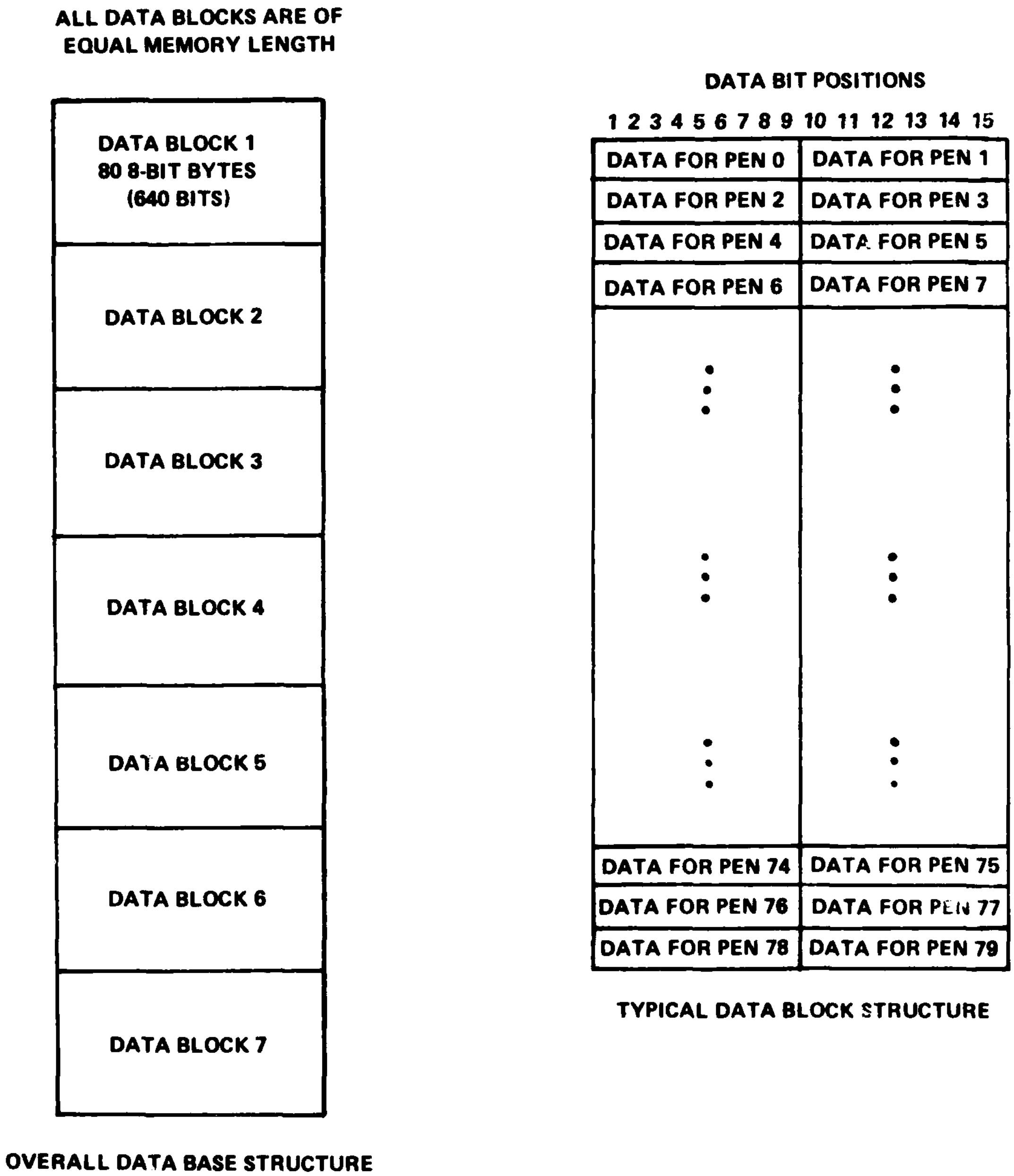


FIG. 19 RECORDER OUTPUT BUFFER DATA STRUCTURE

The execution and operation of the non-resident programs or, for that matter, any other task, in no way interferes with the processing of incoming data from the tactical situation simulator or outputting to the recorder. This aspect is covered under the discussion of operation of the real time operating system.

EDIT Routine (EDIT)

The EDIT Routine is a disc overlay program that enables the operator to create and modify the sensor administration file. Through the teletype, the operator can activate, deactivate, or modify existing sensor records in the file and also activate or deactivate entire receiver channels. Data entered for each sensor in the Administrative File is as follows:

a.	Activate (A) or Deactivate (D) status	ST
b.	Sensor Channel and Identification Number	RID
c.	Recorder Number and Pen Position	RPP
d.	Type of Sensor	TYP
e.	East and North UTM coordinates	EAST-NORTH
f.	Array Number	AR
g.	Date Sensor was Deployed	DDPD

When the EDIT Routine is run to completion, a new active sensor file is created in core (located in the common data base) and the entire sensor file is transferred to a disc. This file contains all active and deactivated sensors. (A deactivated sensor is one which is invisible to the system for processing.) The channel indicator will be set to indicate the status of the receiver channels, which will be set by the input processing task (Task 3). The EDIT Routine also calculates the coordinate scale from the UTM coordinates of all the active sensors and scales of the CRT display appropriately for the mapping of the sensor field.

The Administrative Routine (ADMIN)

The ADMINistrative Routine is a disc overlay program that allows the operator to list on a peripheral device either the entire or any portion of the sensor administrative file on disc. This allows the operator to display the data for any sensor or groups of sensors to any of the peripheral devices (CRT, line printer or teletype). The operator has the following display modes of operations for listing sensors:

1. All sensors in the Administrative File

2. All active sensors in the Administrative File

3. Any particular sensor or group of sensors

4. All sensors within specified UTM coordinates

5. All active sensors within specified UTM coordinates

When the CRT is selected as the output device, (Figure 20), the sensors are listed on two sides of the screen, split screen fashion. This permits a maximum of 43 sensors to be displayed at any time. If there are more than 43, the routine will queue the operator if the rest are to be displayed. When the line printer is selected as the output device (Figure 21), a heading is printed and the list outputted. When sensor activations enter the system during this mode, they are processed to completion immediately, and the activations are listed to the line printer upon completion of outputting the Administrative List.

ACTIVation Routine

The sensor ACTIVation simulation routine is a disc overlay program which generates sensor activations and enters them into the system. This input is the same as if an actual activation occurred at the receiver inputs. Thus, the simulation of any sensor report can be easily accomplished for software testing, demonstrations, et cetera.

The program can produce either classification or detection-only reports. To enter the data one needs only the RID (Receiver - ID) number and the desired classification (if a classifier).

For multiple activations, one has to enter the data separated by commas. This will time-tag all entered data at the identical time of entry. An example of program operation is as follows:

Command	Explanation
ACTIV	Call Program
**	Program Ready
121	Output Activation on Sensor 121
**	Program Ready
131-2	Output Activation and Classification 2
121, 131-2, 121, 141	Output many Activations
**	Program Ready
END	END Program
END ACTIV	Normal Program End.

RID	TYP	EAST	NORTH	RPP	AR	ST	DDPD
101	SFE	04000	50000	111	00	A	1019
102	SFE	04000	49500	112	00	A	1019
103	SFE	04000	49000	113	00	A	1019
104	SFE	04000	44000	116	00	A	1019
105	SFE	04000	43500	117	00	A	1019
106	SFE	04000	43000	118	00	A	1019
107	SFE	04565	42935	119	00	A	1019
108	SFE	04920	42581	120	00	A	1019
109	SFE	04000	38000	123	00	A	1019
111	SFE	04000	36700	125	00	A	1019
112	SFE	04000	32000	130	00	A	1019
113	SFE	04000	31500	131	00	A	1019
114	SFE	04000	31000	132	00	A	1019
115	SFE	04000	30500	133	00	A	1019
164	SFE	04000	37350	124	00	A	1019
225	SFE	21000	48000	150	00	A	1019
226	SFE	21000	47350	151	00	A	1019
227	SFE	21000	46700	152	00	A	1019
228	SFE	21000	37500	154	00	A	1019
229	SFE	21000	37000	155	00	A	1019
231	SFE	20500	37000	157	00	A	1019
232	SFE	20000	37000	158	00	A	1019

RID	TYP	EAST	NORTH	RPP	AR	ST	DDPD
233	SFE	21000	32500	160	00	A	1019
234	SFE	21000	32000	161	00	A	1019
235	SFE	21000	31500	162	00	A	1019
236	SFE	21000	31000	163	00	A	1019
264	SFE	21000	36500	156	00	A	1019
537	SFE	12250	33750	166	00	A	1019
538	SFE	13000	33750	167	00	A	1019
539	SFE	13750	33750	168	00	A	1019
541	SFE	13000	33000	170	00	A	1019
542	SFE	13750	33000	171	00	A	1019
543	SFE	12250	32250	172	00	A	1019
544	SFE	13000	32250	173	00	A	1019
545	SFE	13750	32250	174	00	A	1019
564	SFE	12250	33000	169	00	A	1019
616	SFE	19500	57000	140	00	A	1019
617	SFE	20250	57000	141	00	A	1019
618	SFE	21000	57000	142	00	A	1019
619	SFE	19500	56250	143	00	A	1019
621	SFE	21000	56250	145	00	A	1019
622	SFE	19500	55000	146	00	A	1019
623	SFE	20250	55000	147	00	A	1019

'PAGE COMPLETED'

FIGURE 20. TYPICAL ADMINISTRATIVE FILE CRT OUTPUT

* ADMINISTRATIVE FILE *

RID	TYP	EAST	NORTH	RPP	AR	ST	DDPD
101	SFE	04000	50000	111	00	A	1019
102	SFE	04000	49500	112	00	A	1019
103	SFE	04000	49000	113	00	A	1019
104	SFE	04000	44000	116	00	A	1019
105	SFE	04000	43500	117	00	A	1019
106	SFE	04000	43000	118	00	A	1019
107	SFE	04565	42935	119	00	A	1019
108	SFE	04920	42581	120	00	A	1019
109	SFE	04000	38000	123	00	A	1019
111	SFE	04000	36700	125	00	A	1019
112	SFE	04000	32000	130	00	A	1019
113	SFE	04000	31500	131	00	A	1019
114	SFE	04000	31000	132	00	A	1019
115	SFE	04000	30500	133	00	A	1019
164	SFE	04000	37350	124	00	A	1019
225	SFE	21000	48000	150	00	A	1019
226	SFE	21000	47350	151	00	A	1019
227	SFE	21000	46700	152	00	A	1019
228	SFE	21000	37500	154	00	A	1019
229	SFE	21000	37000	155	00	A	1019
231	SFE	20500	37000	157	00	A	1019
232	SFE	20000	37000	158	00	A	1019
RID	TYP	EAST	NORTH	RPF	AR	ST	DDFD
233	SFE	21000	32500	160	00	A	1019
234	SFE	21000	32000	161	00	A	1019
235	SFE	21000	31500	162	00	A	1019
236	SFE	21000	31000	163	00	A	1019
264	SFE	21000	36500	156	00	A	1019
537	SFE	12250	33750	166	00	A	1019
538	SFE	13000	33750	167	00	A	1019
539	SFE	13750	33750	168	00	A	1019
541	SFE	13000	33000	170	00	A	1019
542	SFE	13750	33000	171	00	A	1019
543	SFE	12250	32250	172	00	A	1019
544	SFE	13000	32250	173	00	A	1019
545	SFE	13750	32250	174	00	A	1019
564	SFE	12250	33000	169	00	A	1019
616	SFE	19500	57000	140	00	A	1019
617	SFE	20250	57000	141	00	A	1019
618	SFE	21000	57000	142	00	A	1019
619	SFE	19500	56250	143	00	A	1019
621	SFE	21000	56250	145	00	A	1019
622	SFE	19500	55000	146	00	A	1019
623	SFE	20250	55000	147	00	A	1019
624	SFE	21000	55000	148	00	A	1019
RID	TYP	EAST	NORTH	RFF	AR	ST	DDFD
664	SFE	20250	56250	144	00	A	1019

* ADMIN FILE COMPLETE *

FIGURE 21. TYPICAL ADMINISTRATIVE FILE LINE PRINTER OUTPUT

Scenario EDIT Routine

The SCenario EDITor performs three functions:

a. Read, check and pass geometry, object, or sensor data to Task 6.

b. Check for current valid data files in Task 6 and output the values to the operator.

c. Execute the Task 6 Tactical Situation Simulator algorithm.

1. Enter Data. Data entry to the algorithm is done by means of prepunched paper tapes which have the desired geometry, object, or sensor data to be used. The program uses a specially written driver on high speed paper tape reader/punch to read in the data tapes. When entering data, all initialization required for the specific data files is performed as well as format, sync, and data record checks.

2. Check Data. The previously entered tape numbers for geometry, object, and sensor data are checked and outputted to the operator.

3. Execute. All pertinent system and Task 6 data are initialized. The proper flags are then set and the program is exited to allow the real time operating system (RTOS) to start the Task 6 tactical situation simulator algorithm.

TASK 5 - Command Directory Task

The function of Task 5 is to read from disc, a directory file of valid names used to call the non-resident programs. Once the list is read into memory, the operator may select one of the non-resident programs to be implemented. This is accomplished by the operator entering the program's name (Example: EDIT). Task 5 searches the directory for the program, EDIT, and its corresponding location on the non-resident disc file. Once the program and location have been found, Task 5 loads the appropriate program, in this example, EDIT, into the operator processing task (Task 4) overlay area. After the program is loaded, Task 5 starts Task 4, thereby giving control to the operator who can now use the non-resident program. In the event an operator enters the name of a non-resident program which does not exist in the directory, an error message will be logged to the teletype by Task 5.

TASK 6 - Tactical Situation Simulator Algorithm Operation

The Task 6 algorithm is started by a non-resident program (SCEDIT) which also edits and checks data (Figure 22). Once started, it initializes several parameters in the data and pointer files, sets CRT constants and several general operating register values. Then it proceeds to a time check routine. Here, the algorithm time is generated by comparison to the real time clock interrupt count. If the required count has not elapsed, the program waits and returns to the scheduler. Once the required count has elapsed (from the previous time, approximately 500 milliseconds), the current state of the system is updated.

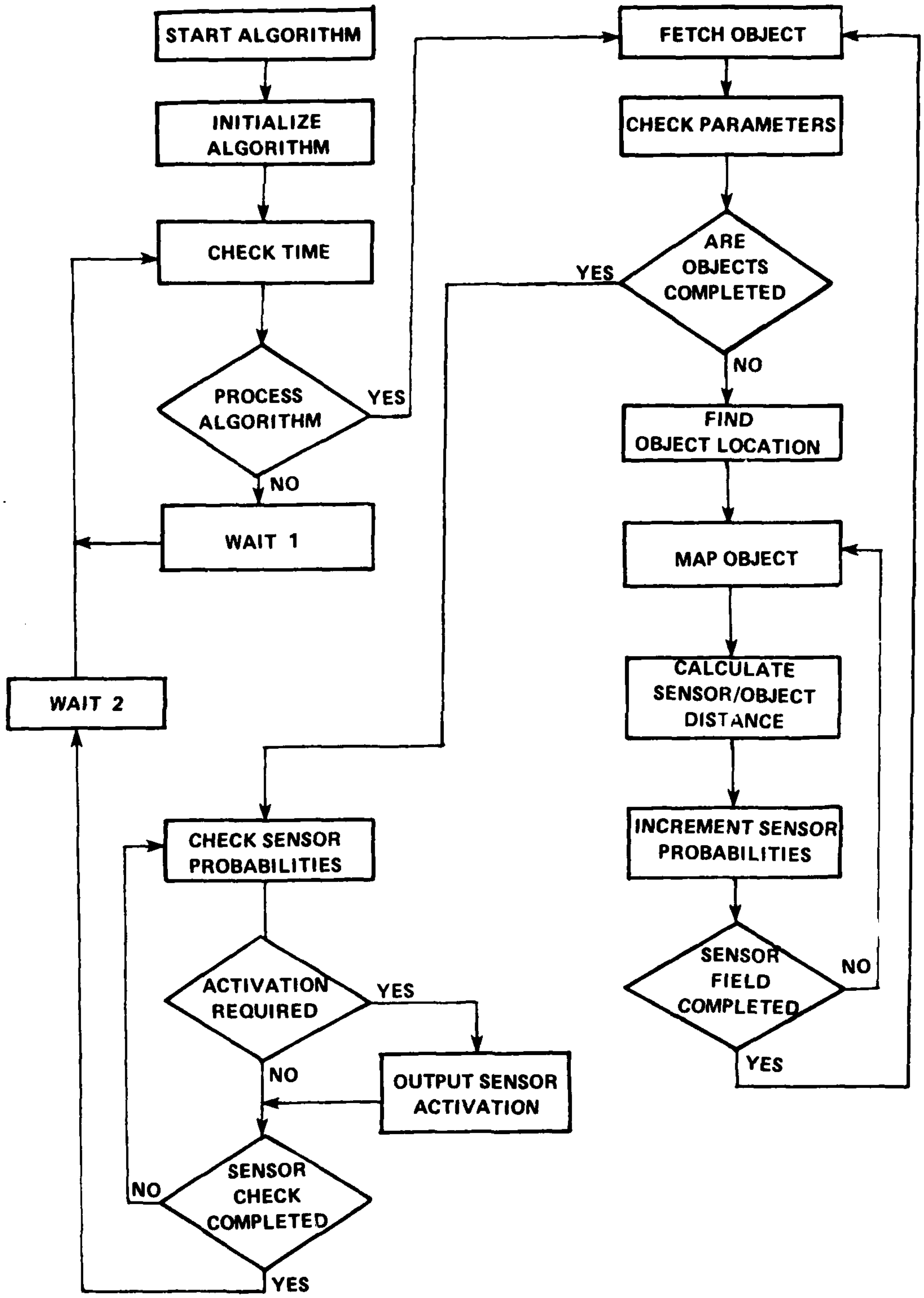


FIG. 22 TASK 6 SIMULATION PROGRAM FLOW DIAGRAM

The system update relocates each object to its new position based on the exact inhibit time value for the current algorithm period, the object speed, and the track which the object is following. The distance between each object and sensor is calculated, and the resulting probabilities and classifications for each sensor are modified, if required.

Once all the objects have been checked against each sensor, the program looks for activation threshold conditions of each sensor. An activation will result whenever the summed probability of detection for a given sensor exceeds a threshold value. Classification is determined from the dominant target which perturbed the given sensor for that period.

After all the sensors have been checked as to their activation condition, the algorithm passes the activation data to Task 3, starts it, and goes into the wait state. When it is finished waiting, it is reactivated and the entire process starts all over again.

The end of the program execution occurs when all the objects reach the end of their respective tracks. At the end, a message is printed to alert the operator to the fact.

General Description

The algorithm basically functions as a discrete state system.[1] A general expression of the describing state equations is as follows:

$$O_i = f_1 \ (G, O_{i-1}) \quad (1) \qquad\qquad (B1)$$

$$S_i = f_2 \ (S_{i-1}, \ O_i) \quad (2) \qquad\qquad (B2)$$

where O is the object state matrix, S is the sensor matrix, G the geometry functions, and i the current state. The f_1 and f_2 are state transition functions.

The current object state O_i is functionally determined in (B1) by the geometry function G and the previous object state O_{i-1}. Then the current sensor state S_i is determined in (B2) from O_i and S_{i-1}, the previous sensor state. S_i is interpreted and the appropriate actions taken until the next operation (B1) is begun. Note that the operations of f_1 and f_2 are done in discrete time intervals, which can be varied depending on the computer speed.

The three state variables which define the system are described by (1) geometry (track) parameters, (2) object parameters, and (3) sensor parameters. The basic characteristics of each are defined before the execution of the program, although certain items may be changed during

1 A discrete state system is a collection of state variables, the value of which at any instant of time, determines the state, or output, of the system.

execution. Thus, the operator merely initializes the state equations with data from (1), (2), and (3) above and starts the program. Once initialized with all three sets of data, the system does not need to be reinitialized unless it is desired to change one or more of the data groups.

APPENDIX C

TASK 1 - THERMAL RECORDER TASK


```
                  *
                  * AUTHOR:  AL SLUTSKY
                  *
                  *
                  *      THMRDR TASK BLOCK
                  *
                  *
                  *
0000R                    ENTRY  RDRTB,RPTR,WPTR,THMFLG,THMRDR
0000R                    ENTRY  TASK1
0000R                    EXTRN  LASTBF,RDRBUF,TEXRUN
                  *
                  *
                  *
0000R             TASK1   EQU   *
0000R             RDRTB   DS    8             UNUSED
0008R 0000                DC    0             PARAMETER
000AR                     DS    2             UNUSED
000CR 0000                DC    0             NO TELL BUFFER
000ER 0000                DC    0             LU 0
0010R 0086                DC    X'86',0,0     LU 1-3
      0000
      0000
0016R 0000                DC    0,0,0,0       LU 4-7
      0000
      0000
      0000
001ER 0000                DC    0,0,0,0       LU 8-11
      0000
      0000
      0000
0026R 0000                DC    0,0,0,0       LU 12-15
      0000
      0000
      0000
002ER                     DS    32            REG SAVE AREA
                  *
                  *
0001              PTR     EQU   1
0001              R1      EQU   1
0002              R2      EQU   2
0003              R3      EQU   3
0000              R0      EQU   0
0006              R6      EQU   6
                  *
                  *
                  *THIS IS THE THERMAL RECORDER TASK(THMRDR).
                  *IT IS MADE READY BY THE THMRDR ISR
004ER             THMRDR  EQU   *
004ER 4810                LH    R1,THMFLG     FIRST TIME THRU TASK?
      011AR
0052R 4330                BZ    INITAL        YES, BRANCH
      009CR
0056R 0711        TEXCHK  XHR   R1,R1         WAIT FOR
0058R 4510                CLH   R1,TEXRUN     TASK 3 TO COMPLETE
      0000F
```

```
005CR 4330            BZ    GO            OUTPUTTING
      0068R
0060R E120            SVC   2,WAIT        CHARACTER
      011ER
0064R 4300            B     TEXCHK
      0056R
0068R 4810      GO    LH    PTR,RPTR      LOAD READ POINTER
      0116R
006CR 4010            STH   PTR,WPTR      SET EQUAL TO WRITE POINTER
      0118R
0070R 4010            STH   PTR,BUFST     STORE INTO PARA  BLOCK
      0112R
0074R CA10            AHI   PTR,79        INC  TO END OF BUFFER
      004F
0078R 4010            STH   PTR,BUFEND    STORE ENDING BUFFER ADDRESS
      0114R
007CR 2611            AIS   PTR,1         INCR  TO NEXT BUFFER
007ER 4510            CLH   PTR,FINAL     ARE WE PAST LAST BUFFER?
      011CR
0082R 2133            BNES  STORE         IF NOT, STORE IT
0084R C810            LHI   PTR,RDRBUF    ELSE, RE-INITALIZE POINTER
      0000F
0088R 4010      STORE STH   PTR,RPTR      STORE POINTER FOR TASK 3
      0116R
008CR E110      WRITE SVC   1,WRIBUF      WRITE TO RECORDER
      010ER
0090R 4800            LH    R0,STAT       GET STATUS IN R0
      0110R
0094R 4230            BNZ   ERROR         IF NOT 0, BRANCH TO ERROR ROUTINE
      00ECR
0098R E130      EOJ   SVC   3,0
      0000
009CR C830      INITAL LHI  R3,LASTBF     LOAD LAST BUFFER ADDRESS
      0000F
00A0R CA30            AHI   R3,80         INCR  TO BUFFER AREA END
      0050
00A4R 4030            STH   R3,FINAL      STORE FINAL BUFFER LOCATION
      011CR
00A8R C810            LHI   R1,RDRBUF     SET RPTR TO START
      0088R
00ACR 4010            STH   R1,BUFST
      0112R
00B0R 4010            STH   R1,RPTR       OF BUFFER
      0116R
00B4R CA10            AHI   R1,79         INCR  TO END OF BUFFER
      004F
00B8R 4010            STH   R1,BUFEND     STORE BUFFER ENDING ADDRESS
      0114R
00BCR E110            SVC   1,WRIBUF
      010ER
00C0R 4800            LH    R0,STAT
      0110R
00C4R 4230            BNZ   ERROR
      00ECR
00C8R C810            LHI   R1,RDRBUF     SET UP BUFFER
      00AAR
```

```
00CCR 4010              STH    R1,BUFST          ADDRESSES TO OUTPUT
      0112R
00D0R 4010              STH    R1,RPTR
      0116R
00D4R CA10              AHI    R1,79             BLANKS FOR RECORDER
      004F
00D8R 4010              STH    R1,BUFEND         GENERATED INTERRUPT
      0114R
00DCR E110              SVC    1,WRIBUF          GENERATE THE FIRST INTERRUPT
      010ER
00E0R 4800              LH     R0,STAT           FROM RECORDER
      0110R
00E4R 4230              BNZ    ERROR
      00ECR
00E8R 4300              B      EOJ
      0096R
              *
              *
00ECR E120    ERROR     SVC    2,UNPACK          UNPACK ERROR MESSAGE
      010AR
00F0R E120              SVC    2,ERRMSG
      00F8R
00F4R E130              SVC    3,0
      0000
              *
              *
              *
00F8R 0007    ERRMSG    DC     X'0007'
00FAR 000E              DC     14
00FCR 492F              DC     C' I/O ERROR
      4F20
      4552
      524F
      5220
0106R         STATUS    DS     4
              *
              *
010AR 0006    UNPACK    DC     X'0006'
010CR 0106R             DC     STATUS
              *
010ER 3801    WRIBUF    DC     X'3801'           WRITE ASCII
0110R         STAT      DS     2
0112R 00CAR   BUFST     DC     RDRBUF
0114R 0000    BUFEND    DC     0
0116R 0112R   RPTR      DC     RDRBUF
0118R 0000    WPTR      DC     0
011AR 0000    THMFLG    DC     0
011CR 0000    FINAL     DC     0
              *
011ER 000B    WAIT      DC     X'000B'
0120R 0000              DC     X'0000'
0122R 0001              DC     X'0001'
0124R                   END
```

```
                    *
                    * AUTHOR: AL SLUTSKY
                    *
                    *
                    *
0000R                       ENTRY RDINIT,THMDVR
0000R                       EXTRN ISROVR,TCBRDR,LIOTRM,THMFLG,IOEXIT
0000R                       EXTRN TCBTAB
                    *
                    *
                    *
                    * THIS IS THE DRIVER INITIALIZATION ROUTINE
                    *ISRINT IS ENTERED THROUGH A SINT. THIS ROUTINE ENABLES THE INT
                    *AND SETS UP THE ISR ADDRESS.
                    *
                    *
                    *
0000R               THMDVR  EQU    *
0000R DEE0          ISRINT  OC     DEV,ENHWD         ENABLE INT, HALFWORD MODE
      010AR
0004R 9DE9                  SSR    DEV,STAT          GET STATUS
0006R C390                  THI    STAT,DU           IS DEVICE UNAVAILABLE?
      0001
000AR 2337                  BZS    DEVUAV            YES, BRANCH
000CR 07AA                  XHR    RA,RA             SET CLOCK
000ER 9A6A                  WDR    R6,RA             HIGH
0010R C8F0                  LHI    RF,ISR            SET ADDRS TO ISR ROUTINE
      0034R
0014R 430D                  B      16(DCB)
      0010
0018R C890          DEVUAV  LHI    STAT,X'A000'      SET STATUS TO DU
      A000
001CR 409D                  STH    STAT,38(DCB)      STORE IN DCB
      0026
0020R DFE0                  OC     DEV,DISARM        DISARM INT
      010BR
0024R C8FD                  LHI    RF,20(DCB)        SET ADDRS TO IGNORE INT
      0014
0028R 24D1                  LIS    DCB,1
002AR 64D0                  ATL    DCB,LIOTRM
      0000F
002ER 27D1                  SIS    DCB,1
0030R 430D                  B      16(DCB)           RETURN AND SAVE REGS
      0010
                    *
                    *
                    *
                    * THIS ROUTINE MAKES THE RECORDER TASK (THMRDR) READY WHEN AN IN
                    * IS RECEIVED FROM THE RECORDER SIGNALING THAT IT IS READY TO RE
                    * NEW DATA
                    *
                    *
0034R               ISR     EQU    *
0034R DEF0                  OC     DEV,DISARM        DISARM INTERRUPTS
      010BR
0038R 489C                  LH     STAT,6(TCB)       LOAD TASK STATUS
```

```
          0006
0030R C590                 CLHI   STAT,X'8000      DORMANT
          8000
0040R 423D                 BNE    16(DCB)          NO, RETURN
          0010
0044R 4890                 LH     STAT,18(TCB)     SET
          0012
0048R 4090                 STH    STAT,32(TCB)     CURRENT
          0020
004CR 4890                 LH     STAT,20(TCB)     PSW TO
          0014
0050R 4090                 STH    STAT,34(TCB)     INITIA PSW
          0022
0054R 0B99                 SHR    R9,R9            TASK STATUS
0056R 4090                 STH    STAT,6(TCB)      ZERO R9
          0006
005AR DEE0                 OC     DEV,ENABWD       ENABLE INTERRUPTS
          010AR
005ER 24D1                 LIS    DCB,1
0060R 64D0                 ATL    DCB,LIOTRM
          002CR
0064R 27D1                 SIS    DCB,1
0066R 430D                 B      16(DCB)          RETURN
          0010
                  *
                  *
                  *
                  *THIS IS THE DRIVER FOR THE THERMAL RECORDER. IT IS CALLED BY SV
                  *
                  *
0000              R0       EQU    0
0001              R1       EQU    1
0002              R2       EQU    2
0003              R3       EQU    3
0004              R4       EQU    4
0005              R5       EQU    5
0006              R6       EQU    6
0007              R7       EQU    7
0008              R8       EQU    8
0009              STAT     EQU    9
0009              R9       EQU    9
000A              RA       EQU    10
000A              R10      EQU    10
000B              RB       EQU    11
0001              DU       EQU    1
000C              TCB      EQU    12
000D              DCB      EQU    13
000E              DEV      EQU    14
000F              RF       EQU    15
0038              WRITE    EQU    X'38'
                  *
006AR             RDINIT   EQU    *
006AR 4890                 LH     R9,INGFLG        IF NOT FIRST TIME BRANCH TO
          0094
006ER 4230                 BNZ    RDRDVR           RDRDVR
          00A0R
```

```
0072R 24A1                LIS   R10,1            SET THMFLG
0074R 40A0                STH   R10,THMFLG
      006CR
0078R 08C2                LHR   TCB,R2           LOAD TCB,DCB,DEV
007AR 90C8                SRLS  TCB,8
007CR 48CC                LH    TCB,TCBTAB(TCB)  GET TCB ADDRESS
      0000F
0080R 08D1                LHR   DCB,R1           ADDR FOR EVENTUAL STORAGE
0082R 08E6                LHR   DEV,R6           IN DCB REGS AREA
0084R C8F0                LHI   RF,ISRINT        LOAD ADDR FOR ISRINT
      0000R
0088R C880                LHI   R8,X'4000'
      4000
008CR 9598                EPSR  R9,R8            MASK INT. EXTERNAL
008ER D020                STM   R2,28(DCB)       SAVE REGS FOR ISRINT
      001C
0092R E20E                SINT  0(DEV)           GO TO ISRINT ROUTINE
      0000
0096R 9589                EPSR  R8,R9
0098R 487D                LH    R7,38(DCB)       LOAD STATUS FROM DCB
      0026
009CR 4300                B     IOEXIT           EXIT DRIVER INITIALIZE
      0000F
                *
                *
                *
00A0R 08F4      RDRDVR    LHR   R8,R4            LOAD FUNCT AND LU
00A2R 9088                SRLS  R8,8             ISOLATE FUNCTION
00A4R C580                CLHI  R8,WRITE         IS IT A WRITE
      0038
00A8R 4230                BNE   ILFUNC           NO, BRANCH
      00B8R
00ACR 9D69                SSR   R6,STAT          GET STATUS
00AER C390                THI   STAT,DU          IS DEVICE UNAVAIABLE
      0001
00B2R 4230                BNZ   DEVNAV           YES BRANCH
      00C0R
00B6R 2309                BS    OUTPUT           BRANCH TO OUTPUT ROUTINE
                *
                *
                *
00B8R C870      ILFUNC    LHI   R7,X'C000'
      C000
00BCR 4300                B     IOEXIT
      009ER
                *
                *
                *
00C0R C870      DEVNAV    LHI   R7,X'A000'       SET STATUS TO DU
      A000
00C4R 4300                B     IOEXIT           BRANCH TO EXIT
      00EER
                *
                *
                *
00C8R 4853      OUTPUT    LH    R5,4(R3)         START ADR OF WRITE BUFFER
```

```
        0004
00CCR 0700             XHR    R0,R0
00CER C8A0             LHI    R10,128          CLOCK IS 7TH BIT OF HALFWORD
        0080
00D2R 48B0             LH     RB,X'2EA'        TO SYNC CLOCK
        02EA
00D6R 45B0    SYNC     CLH    RB,X'2EA'        SO THAT OUTPUTTINY STARTS
        02EA
00DAR 4330             BE     SYNC             AFTER REAL TIME CLK INTERRUPT
        00D6R
00DER DE60             OC     R6,DISARM
        010DR
00E2R 2302             BS     OUT
00E4R 2651    OUT1     AIS    R5,1
00E6R D395    OUT      LB     R9,0(R5)         LOAD BYTE
        0000
00EAR C790             XHI    R9,X'1F'         COMPEMENT FOR OUTPUT
        001F
00EER 9A69             WDR    R6,R9            OUTPUT CHAR
00F0R 069A             OHR    R9,R10           AND CLOCK TO CHAR
00F2R 9A69             WDR    R6,R9            OUTPUT BYTE PLUS CLOCK
00F4R D205             STB    R0,0(R5)         ZERO BUF LOC FROM WHICH CHAR WAS O
        0000
00F8R 4553             CLH    R5,6(R3)         IF 80 CHAR HAVE NOT BEEN OUTPUT
        0006
00FCR 4230             BNE    OUT1             BRANCH TO OUT1
        00E4R
0100R 0777             XHR    R7,R7
0102R DE60             OC     R6,ENHWD
        010AR
0106R 4300             B      IOEXIT           RETURN TO USER
        00C4R
010AR 6000    ENHWD    DC     X'6000'
010BR         DISARM   EQU    ENHWD+1
010CR                  END
```

TASK 2

LINE PRINTER TASK


```
                   *
                   *
                                * LINE   PRINTER   TASK
                   *
                   *   REWRITTEN  AND  DEBUGGED  BY
                   *   RICH MARTINO & JOE KARAKOWSKI
                   *
                   *
                   *
0000R                     EXTRN LPFLAG, ININUM
0000R                     EXTRN CURSEC, CURNUM, INISEC, DBUFF1, DBUFF2
0000R                     EXTRN @AHLOX, @AHLOY, @AHRN, @AHSZE
0000R                     EXTRN @AHTBR, @AHTIM, @AHTYP
0000R                     ENTRY LNP, LNPTB
                   *
                   * LINE PRINTER TASK CONTROL BLOCK
                   *
0000R              LNPTB  DS    8                 UNUSED
0008R 0000                DC    0                 PARAMETER
000AR                     DS    2                 UNUSED
000CR 0000                DC    0                 NO TELL BUFFER
000ER 0000                DC    0,X'62',X'406',0 LU 0-3
      0062
      0406
      0000
0016R 0000                DC    0,0,0,0           LU 4-7
      0000
      0000
      0000
001ER 0000                DC    0,0,0,0           LU 8-11
      0000
      0000
      0000
0026R 0000                DC    0,0,0,0           LU 12-15
      0000
      0000
      0000
002ER                     DS    32                REGISTER SAVE AREA
                   *
0000               AHRN   EQU   0
0004               AHTYP  EQU   4
0009               AHLOX  EQU   9
0010               AHLOY  EQU   16
0017               AHTIM  EQU   23
001E               AHTBR  EQU   30
                   *
0007               NOPGE  EQU   7
0001               CSEC   EQU   1
0002               CNUM   EQU   2
0003               CBUF   EQU   3
0004               BUFA   EQU   4
000A               R10    EQU   10
000B               R11    EQU   11
000C               R12    EQU   12
000D               R13    EQU   13
000E               R14    EQU   14
```

```
000F            R15     EQU    15
0009            WORK    EQU    9
0005            RS      EQU    5
                *
0000            ZERO    EQU    0
001F            PGEMAX  EQU    30
0008            CMAX    EQU    8
9006            EOF     EQU    X'9006'
                *
                *
                *  LINE PRINTER TASK  MAIN PROGRAM
                *
                *
                *
004ER           LNP     EQU    *
004ER 0B00              SHR    ZERO,ZERO          SET REG
0050R 4000              STH    ZERO,IOFLG         SET FLAG TO NO ERROR
      01F8R
0054R 4810              LH     CSEC,INISEC        LOAD INITIAL SECTOR OF THE PROGRA
      0000F
0058R 4820              LH     CNUM,ININUM
      0000F
005CR C520              CLHI   CNUM,CMAX
      0008
0060R 4280              BL     LDBUF
      0066R
0064R 0B22      REINIT  SHR    CNUM,CNUM          0 CURRENT NUMBER
0066R C830      LDBUF   LHI    CBUF,DBUFF1        LOAD CURRENT BUFFER ADDR
      0000F
006AR C310              THI    CSEC,1
      0001
006ER 2333              BZS    DONE
0070R C830              LHI    CBUF,DBUFF2
      0000F
0074R           DONE    EQU    *
0074R 4890      CHKSEC  LH     WORK,LPFLAG        LOAD L. P. FLAG INTO REG
      0000F
0078R 4210              BM     CHKFLG
      0088R
007CR C870              LHI    NOPGE,PGEMAX       LOAD REG WITH NO. ON PAGE
      001E
0080R C890              LHI    WORK,-1            LOAD LINE PRINTER FLAG WITH -1
      FFFF
0084R 4090              STH    WORK,LPFLAG
      0076R
0088R 2129      CHKFLG  BPS    WAIT               BRANCH IF POSITIVE
008AR 4510              CLH    CSEC,CURSEC        SECTORS EQUAL ?
      0000F
008ER 4230              BNE    PAGE2
      0146R
0092R 4520              CLH    CNUM,CURNUM        RECORD NUMBERS EQUAL
      0000F
0096R 4230              BNE    GETNXT
      00A2R
009AR           WAIT    EQU    *
009AR E120              SVC    2,INTPBK
```

```
       0200R
009ER 4300              B      CHKSEC
       0074R
00A2R 0843     GETNXT   LHR    BUFA,CBUF          ADD TO NUMBER SINCE LAST PAGE
00A4R 2671     NEXTRC   AIS    NOPGE,1
00A6R C570              CLHI   NOPGE,PGEMAX       IF LESS NO HEADING
       001E
00AAR 4320              BNP    SKPHD
       00C0R
00AER 0B77              SHR    NOPGE,NOPGE
00B0R          HDSVC    EQU    *
00B0R E110              SVC    1,HDRBLK           WRITE ASCII AND WAIT ON LU 1
       0212R
00B4R 4850              LH     R5,HDRBLK+2        LOAD STATUS
       0214R
00B8R 4230              BNZ    IOERR1             IF ERROR BRANCH
       0180R
00BCR 4000              STH    ZERO,IOFLG         RESET FLAG
       01F8R
00C0R 08B2     SKPHD    LHR    R11,CNUM
00C2R C8A0              LHI    R10,@AHSZE
       000F
               *
               * LINE PRINTER BUFFER FORMATING
               *
00C6R 0CAA              MHR    R10,R10            COMPUTE DISP INTO BUFFER
00C8R 0A4B              AHR    BUFA,R11           GET ADRS OF DISC RECORD
               * FORMAT RID ( RID )
00CAR 2403              LIS    R12,3              LENGTH
00CCR C8D4              LHI    R13,@AHRN(BUFA)    ORGIN
       000F
00D0R C8E0              LHI    R14,LPBUF+AHRN
       023CR
00D4R 41F0              BAL    R15,MVC
       0266R
               * FORMAT TYPE ( TTT )
00D8R 2403              LIS    R12,3
00DAR C8D4              LHI    R13,@AHTYP(BUFA)
       000F
00DER C8E0              LHI    R14,AHTYP+LPBUF
       0240R
00E2R 41F0              BAL    R15,MVC
       0266R
               * FORMAT EAST ( XXXXXX )
00E6R 2406              LIS    R12,6
00E8R C8D4              LHI    R13,@AHLOX(BUFA)
       000F
00ECR C8E0              LHI    R14,LPBUF+AHLOX
       0245R
00F0R 41F0              BAL    R15,MVC
       0266R
               * FORMAT NORTH ( YYYYYY )
00F4R 2406              LIS    R12,6
00F6R C8D4              LHI    R13,@AHLOY(BUFA)
       000F
00FAR C8E0              LHI    R14,LPBUF+AHLOY
```

```
       024CR
00FER 41F0                BAL   R15,MVC
       0266R
              * FORMAT TIME ( HHMMSS )
0102R 2406                LIS   R12,6
0104R C8D4                LHI   R13,@AHTIM(BUFA)
       0000F
0108R C8E0                LHI   R14,LPBUF+AHTIM
       0253R
010CR 41F0                BAL   R15,MVC
       0266R
              * FORMAT TBR ( **** )
0110R 24C4                LIS   R12,4
0112R C8D4                LHI   R13,@AHTBR(BUFA)
       0000F
0116R C8E0                LHI   R14,LPBUF+AHTBR
       025AR
011AR 41F0                BAL   R15,MVC
       0266R
011ER         LPSVC       EQU   *
011ER E110                SVC   1,LPBLK            WRITE ASCII AND WAIT ON LU 2
       025ER
0122R 4850                LH    R5,LPBLK+2
       0260R
0126R 4230                BNZ   IOERR2
       0188R
012AR 4000                STH   ZERO,IOFLG         SET FLAG TO NO ERROR
       01F8R
012ER 2621                AIS   CNUM,1
0130R 4020                STH   CNUM,ININUM
       005AR
0134R C520                CLHI  CNUM,CMAX
       0008
0138R 4280                BL    CHKSEC
       0074R
013CR 2611                AIS   CSEC,1
013ER 4010                STH   CSEC,INISEC
       0056R
0142R 4300                B     REINIT             BRANCH TO REINITALIZE
       0064R
0146R C8F1    PAGE2       LHI   R15,1(CSEC)
       0001
014AR 45F0                CLH   R15,CURSEC
       008CR
014ER 4330                BE    GETNXT
       00A2R
0152R C840                LHI   BUFA,DSCBUF
       027AR
0156R 4510                CLH   CSEC,DFBLK+8
       0382R
015AR 4330                BE    NEXTRC
       00A4R
015ER 4010    DOSVC       STH   CSEC,DFBLK+8
       0382R
0162R         DSCSVC      EQU   *
0162R E110                SVC   1,DFBLK            WRITE RANDOM AND WAIT ON LU 2
```

```
        037AR
0166R 4850              LH    R5,DPBLK+2        LOAD STATUS
        037CR
016AR 4330              BZ    NEXTRC
        00A4R
016ER C550              CLHI  R5,EOF
        9006
0172R 4230              BNE   IOERR3
        0190R
0176R 4000              STH   ZERO,IOFLG
        01F8R
017AR 0B11              SHR   CSEC,CSEC
017CR 4300              B     DOSVC
        015ER
0180R 41F0    IOERR1    BAL   R15,IOERR
        0198R
0184R 4300              B     HDSVC             BRANCH TO PRINT HEADING AGAIN
        00B0R
0188R 41F0    IOERR2    BAL   R15,IOERR
        0198R
018CR 4300              B     LPSVC             BRANCH TO PRINT LNPTR BUFFER AGAIN
        011ER
0190R 41F0    IOERR3    BAL   R15,IOERR
        0198R
0194R 4300              B     DSCSVC            BRANCH TO DISC WRITE AGAIN
        0162R
0198R 4890    IOERR     LH    WORK,IOFLG
        01F8R
019CR 4230              BNZ   WAIT2
        01B4R
01A0R 40F0              STH   R15,IOFLG
        01F8R
01A4R 4050              STH   R5,ERSTAT
        0208R
01A8R 41F0              BAL   R15,UNPK          BRANCH TO HEX/ASCII UNPACK ROUTINE
        01BER
01ACR 0208R             DC    ERSTAT,STATUS
        0208R
01B0R E120              SVC   2,ERROR
        01FAR
01B4R E120    WAIT2     SVC   2,INTFBK
        020CR
01B8R 48F0              LH    R15,IOFLG
        01F8R
01BCR 030F              BR    R15
              *
              * UNPACK ROUTINE TO CNVERT HEX TO ASCII
              *
01BER 48EF    UNPK      LH    R14,0(R15)
        0000
01C2R 48DF              LH    R13,2(R15)
        0002
01C6R 24CC              LIS   R12,12
01C8R 48BE              LH    R11,0(R14)
        0000
01CCR 08AB    UNPK1     LHR   R10,R11
```

```
01CER DCAC           SRHL  R10,0(R12)
      0000
01D2R C4A0           NHI   R10,15
      000F
01D6R D3AA           LB    R10,UTAB(R10)
      01E8R
01DAR D2AD           STB   R10,0(R13)
      0000
01DER 26D1           AIS   R13,1
01E0R 27C4           SIS   R12,4
01E2R 221B           BNMS  UNPK1
01E4R 430F           B     4(R15)
      0004
                *
                * PARAMETER BLOCKS
                *
01E8R 3031      UTAB     DC    C'0123456789ABCDEF'
      3233
      3435
      3637
      3839
      4142
      4344
      4546
01F8R           IOFLG    DS    2                   I/O ERROR FLAG
01FAR 0007      ERROR    DC    7,14,C'I/O ERROR '
      000E
      492F
      4F20
      4552
      524F
      5220
0208R           STATUS   DS    4
0208R           ERSTAT   EQU   STATUS
020CR 000B      INTPBK   DC    11,0,1000           ONE SECOND WAIT
      0000
      03E8
0212R 2801      HDRBLK   DC    X'2801',0,BEGHDR,ENDHDR
      0000
      021AR
      0258R
021AR 5249      BEGHDR   DC    C'RID TYPE  EAST  NORTH   TIME  TBR
      4420
      5459
      5045
      2020
      4541
      5354
      2020
      4E4F
      5254
      4820
      2020
      5449
      4D45
      2020
```

```
        5442
        5220
023BR                ENDHDR   EQU    *-1
023CR   5249         LPBUF    DC     C'RID TTT  XXXXXX YYYYYY HHMMSS TBR '
        4420
        5454
        5420
        2058
        5858
        5858
        5820
        5959
        5959
        5959
        2048
        484D
        4D53
        5320
        5442
        5270
025DR                LPBUFE   EQU    *-1
025ER   2801         LPBLK    DC     X'2801',0,LPBUF,LPBUFE
        0000
        023CR
        025DR
                     *
                     * BYTE HANDLING ROUTINE FOR EACH FORMAT
                     *
0266R                MVC      EQU    *
0266R   D39D         NXTB     LB     WORK,0(R13)
        0000
026AR   D29E                  STB    WORK,0(R14)
        0000
026ER   26D1                  AIS    R13,1
0270R   26E1                  AIS    R14,1
0272R   27C1                  SIS    R12,1
0274R   4270                  BP     NXTB              BRANCH ON PLUS TO NEXT BYTE
        0266R
0278R   030F                  BR     R15               RETURN
027AR                DSCBUF   DS     256
037AR   4C02         DPBLK    DC     X'4C02',0,DSCBUF,DSCBUF+255,-1
        0000
        027AR
        0379R
        FFFF
0384R                         END
```

TASK 3

INPUT PROCESSING TASK


```
                 *
                 *
                 *     INPUT  PROCESSING  TASK
                 *
                 *
                 *     REWRITTEN  AND  DEBUGGED  BY .
                 *    RICH MARTINO  &  JOE KARAKOWSKI
                 *
                 *
                 *
                 * TO INITIALIZE THE INPUT PROCESSING ROUTINE, THE BITS
                 * OF NCHSTA MUST FIRST BE SET FOR EACH RECVR CHANNEL
                 * TO BE ACTIVATED (NUMBERED FROM RIGHT  TO LEFT).
                 * THEN A CALL TASK MESSAGE (CHMESS) OF "0" MUST BE
                 * PLACED IN THE INPUT QUEUE (INPQ).
                 *
                 * THE RECEIVER LOGICAL UNIT NUMBER IS THE SAME AS THE RECEIVER
                 * CHANNEL NUMBER
                 *
                 *  DEFINED CONSTANTS
                 *
0000             R0       EQU   0
0001             R1       EQU   1
0002             R2       EQU   2
0003             R3       EQU   3
0004             R4       EQU   4
0005             R5       EQU   5
0006             R6       EQU   6
0007             R7       EQU   7
0008             R8       EQU   8
0009             R9       EQU   9
000A             R10      EQU   10
000B             R11      EQU   11
000C             R12      EQU   12
000D             R13      EQU   13
000E             R14      EQU   14
000F             R15      EQU   15
0020             RST      EQU   X'20'
0040             SET      EQU   X'40'
000E             LINK1    EQU   14
000D             LINK2    EQU   13
0022             ABSZE    EQU   34
                 *
0001             RCV1     EQU   1                 RCVR LOGICAL UNITS
0002             RCV2     EQU   2
0003             RCV3     EQU   3
0004             RCV4     EQU   4
0005             RCV5     EQU   5
0006             RCV6     EQU   6
                 *
0006             NOCHAN   EQU   6                 NO. OF RECVR CHAN'S
                 *
0000R                     ENTRY INPQ, INPPRO, TASK3, NUMRCH, CHMESS
0000R                     ENTRY RCVR1, INTBUF
                 *
0000R                     EXTRN ACTADM, ADMTAB, CHSTA, NCHSTA
```

```
0000R                    EXTRN CONASC,CRTOUT,TYPE
0000R                    EXTRN INCTSE,WDCTSE,ACCTSE
0000R                    EXTRN INCTSI,WDCTSI,ACCTSI
0000R                    EXTRN @ADMSZ,@LTMAC
0000R                    EXTRN @LTMA2
0000R                    EXTRN @NOACT,@SNSTP,@LCBR
0000R                    EXTRN TEXAS,@RECNO
               *
               *
               *   INPUT PROCESSING TASK CONTROL BLOCK
               *
               *
0000R          TASK3     DS    8                  UNUSED
0008R                    DS    2                  ADRS INPUT SLOT
000AR                    DS    2                  UNUSED
000CR                    DS    2                  UNUSED TELL BUFFER
000ER 0000               DC    0                  LU00-UNUSED
0010R 0080               DC    X'80'              LU01-RCVR1
0012R 0081               DC    X'81'              LU02-RCVR2
0014R 0082               DC    X'82'              LU03-RCVR3
0016R 0083               DC    X'83'              LU04-RCVR4
0018R 0084               DC    X'84'              LU05-RCVR5
001AR 0085               DC    X'85'              LU06-RCVR6
001CR 0000               DC    0,0
      0000
0020R 0010               DC    X'10'              LU09-SITUATION DISPLAY
0022R 0014               DC    X'14               LU10-AREA DISPLAY
0024R 0000               DC    0
0026R 0000               DC    0,0,0,0            UNUSED LU'S
      0000
      0000
      0000
002ER                    DS    32                 REGISTER SAVE AREA
               *
004ER FC00     INFO      DC    252,0,0,0
      0000
0052R                    DS    504                INPUT QUEUE STORAGE
               *
               *
               * INPUT PROCESSING   MAIN PROGRAM
               *
               *
               *
024AR 0700     INPPRO    XHR   R0,R0              LOAD R0 WITH 0
024CR 2411               LIS   R1,1               LOAD R1 WITH 1
               *
               *
               * INPUT DATA, VALIDATE, AND CHECK RECEIVER STATUS
               *
024ER 6630     DATAIN    RTL   R3,INFO            INPUT CHANNEL NUMBER
      004ER
0252R 4340               BFL   4,CKDATA           IF LST ENTRY, GO TO CKDATA
      0284R
0256R 0722               XHR   R2,R2              RESET ENTRY FLAG
0258R E120     LSTERR    SVC   2,ERRLST           PRINT "NO LIST ENTRY"
      02C4R
```

```
0250R 4300              B       RCSTCK            BRANCH TO RCSTCK
      0322R
                *
                * THE FOLLOWING IS THE DEFINITION OF A RECEIVER PARAMETER
                * BLOCK.
                *
                * THE PARAMETER BLOCK CONTAINS:
                *      FCT CODE (BYTE 1)
                *      LOGICAL UNIT NO. (BYTE 2)
                *      STATUS (BYTE 3)
                *      DEVICE NUMBER (BYTE 4)
                *      INPUT QUEUE NAME (BYTE 5 & 6)
                *
                * THE RECEIVER LOGICAL UNIT NUMBER IS THE SAME AS THE
                * RECEIVER CHANNEL NUMBER
                *
                *      RECEIVER PARAMETER BLOCKS
                *
0260R 4001      RCVR1   DC      X'4000'+RCV1
0262R                   DS      2
0264R 004ER             DC      INFO
                *
0266R 4002      RCVR2   DC      X'4000'+RCV2
0268R                   DS      2
026AR 004ER             DC      INFO
                *
026CR 4003      RCVR3   DC      X'4000'+RCV3
026ER                   DS      2
0270R 004ER             DC      INFO
                *
0272R 4004      RCVR4   DC      X'4000'+RCV4
0274R                   DS      2
0276R 004ER             DC      INFO
                *
0278R 4005      RCVR5   DC      X'4000'+RCV5
027AR                   DS      2
027CR 004ER             DC      INFO
                *
027ER 4006      RCVR6   DC      X'4000'+RCV6
0280R                   DS      2
0282R 004ER             DC      INFO
                *
                *
0284R 2421      CKDATA  LIS     R2,1              SET ENTRY FLAG
0286R 0843              LHR     R4,R3             LOAD R4 WITH DATA
0288R 4310              BNM     CKMESS            IF NOT MINUS, GO TO CKMESS
      02EER
                *
028CR C430      CKSTAT  NHI     R3,X'7FFF'        REMOVE SIGN BIT
      7FFF
0290R 0930              CHR     R3,R0             COMPARE CH NO. WITH 0
0292R 4330              BE      INVDTA            IF 0, GO TO INVDTA
      0300R
0296R C930              CHI     R3,NOCHAN         COMPARE WITH NO. CH
      0006
029AR 4220              BP      INVDTA            IF PLUS, GO TO INVDTA
```

```
        0300R
029ER C883              LHI    R8,X'30'(R3)     LOAD R8 WITH CH NO.  IN ASCII
        0030
02A2R D280              STB    R8,UNAVAL+13     STORE CH NO.  IN MESSAGE
        0315R
02A6R E120              SVC    2,UNAVAL         PRINT "RECVR NO.  X NOT AVAIL "
        0308R
02AAR C840              LHI    R4,X'FFFF'       LOAD R4 WITH X'FFFF'
        FFFF
02AER C850              LHI    R5,X'FFFE'       LOAD R5 WITH X'FFFE'
        FFFE
02B2R EB43              RLL    R4,-1(R3)        SHIFT LEFT NO.  CH - 1
        FFFF
02B6R 4450              NH     R5,CHSTA         RESET CHAN STATUS BIT
        0000F
02BAR 4050              STH    R5,CHSTA         STORE CHSTA WITH BIT RESET
        02B8R
02BER E130              SVC    3,0              EOJ
        0000
              *
02C2R 0000    CHMESS    DC     0                CALL TASK MESSAGE
              *
02C4R 0007    ERRLST    DC     7
02C6R 000E              DC     14
02C8R 4E4F              DC     C'NO LIST ENTRY'
        2040
        4953
        5420
        454E
        5452
        5920
              *
02D6R 0007    ERRDAT    DC     7
02D8R 0014              DC     20
02DAR 494E              DC     C'INVALID CHANNEL NO. '
        5641
        4C49
        4420
        4348
        414E
        4E45
        4C20
        4E4F
        2E20
              *
02EER 2135    CKMESS    BNZS   CKOK             IF NOT 0,  GO TO CKOK
02F0R 48C0              LH     R12,CHSTA        LOAD R12 WITH CHAN STATUS
        02BCR
02F4R 4300              B      STACHK           GO TO STACHK
        032ER
              *
02F8R C930    CKOK      CHI    R3,NOCHAN        COMP CHAN NO  WITH NO  CHAN
        0006
02FCR 4320              BNP    RCSTCK           IF NOT PLUS,  GO TO RCSTCK
        0322R
0300R E120    INVDTA    SVC    2,ERRDAT         PRINT "INVALID DATA"
```

```
     0206R
0304R E130              SVC    3,0                EOJ
     0000
               *
               *
0308R 0007     UNAVAL   DC     7
030AR 0016              DC     22
030CR 5243              DC     C'RCVR NO.    UNAVAILABLE'
     5652
     204E
     4F2E
     2020
     2055
     4E41
     5641
     494C
     4142
     4C45
               *
               *               RECEIVER STATUS ROUTINE
               *
0322R 48C0     RCSTCK   LH     R12,CHSTA          LOAD R12 WITH CHAN STATUS
     02F2R
0326R 49C0              CH     R12,NCHSTA         COMP CHSTA WITH NCHSTA
     0000F
032AR 4330              BE     CKENT              IF 0, SKIP UPDATE
     0436R
               *
032ER 48D0     STACHK   LH     R13,NCHSTA         LOAD R13 WITH NXT CHAN STATUS
     0328R
0332R C841              LHI    R4,NOCHAN(R1)      LOAD R4 WITH NO. CH + 1
     0006
0336R 0851              LHR    R5,R1              LOAD R5 WITH 1
0338R CD54              SLHL   R5,-1(R4)          SHFT R5 LFT BY NO. CH
     FFFF
033CR 9051     NXTCH    SRLS   R5,1               SHFT R5 RGT BY 1 BIT POSITION
033ER 0B41              SHR    R4,R1              SUBTRACT 1 FROM CH NO.
0340R 4330              BZ     PRTMES             IF 0, GO TO PRTMES
     0426R
0344R C894              LHI    R9,-1(R4)          LOAD R9 WITH NO. CH - 1
     FFFF
0348R 4C90              MH     R8,SIX             MULT R9 BY 6
     0360R
034CR 0875              LHR    R7,R5              LOAD R7 WITH SET BIT
034ER 047D              NHR    R7,R13             ISOLATE STATUS BIT
0350R 2336              BZS    RSTCHA             IF 0, GO TO RSTCHA
0352R D1E0              LM     R14,ON             LOAD RCVR STATUS IN ASCII
     0416R
0356R C8B0              LHI    R11,SET            LOAD R11 WITH FCT CODE
     0040
035AR 2305              BS     SETRST             GO TO SETRST
035CR D1E0     RSTCHA   LM     R14,OFF            LOAD RCVR STATUS IN ASCII
     041AR
0360R C8B0              LHI    R11,RST            LOAD R11 WITH FCT CODE
     0020
0364R D2B9     SETRST   STB    R11,RCVR1(R9)      STORE FCT CODE IN PGS BLK
```

```
     0260R
0368R E119             SVC   1,RCVR1(R9)       SET OR RESET RCVR
     0260R
036CR 4889             LH    R8,RCVR1+2(R9)    CHECK STATUS
     0262R
0370R 2136             BNZS  COMPBT            IF NOT 0, GO TO COMPBT
0372R D0E9             STM   R14,RCVSTA+2(R9)  STORE RCVR STATUS IN MESS.
     03F4R
0376R 08A7             LHR   R10,R7            LOAD R10 WITH STATUS BIT
0378R 4300             B     SETBIT            GO TO SETBIT
     03A0R
037CR C480    COMPBT   NHI   R8,X'FF00'        DELETE RIGHT BYTE FROM R8
     FF00
0380R C980             CHI   R8,X'A000'        COMPARE STATUS WITH X'A000'
     A000
0384R 2136             BNZS  ILEG              IF NOT 0, GO TO ILEG
0386R D1E0             LM    R14,DU            LOAD RCVR STATUS IN ASCII
     041ER
038AR D0E9             STM   R14,RCVSTA+2(R9)  STORE RCVR STATUS IN MESS.
     03F4R
038ER 2305             BS    CONTCK            BRANCH TO CONTCK
0390R D1E0    ILEG     LM    R14,ILL           LOAD RCVR STATUS IN ASCII
     0422R
0394R D0E9             STM   R14,RCVSTA+2(R9)  STORE RCVR STATUS IN MESS
     03F4R
0398R C8A0    CONTCK   LHI   R10,X'FFFF'       LOAD R10 WITH ALL 1'S
     FFFF
039CR 0BA7             SHR   R10,R7            COMPLEMENT BIT
039ER 04A5             NHR   R10,R5            AND WITH SET BIT
03A0R C890    SETBIT   LHI   R9,X'FFFF'        LOAD R9 WITH ALL 1'S
     FFFF
03A4R 0B95             SHR   R9,R5             COMPLEMENT SET BIT
03A6R 04C9             NHR   R12,R9            REMOVE BIT FROM CHSTA
03A8R 06CA             OHR   R12,R10           ADD BIT TO CHSTA
03AAR 4300             B     NXTCH             GO TO NXTCH
     033CR
              *
03AER 0006    NUMBCH   DC    NOCHAN            NUMBER OF CHANNELS
03B0R 0006    SIX      DC    6
              *
03B2R 0007    RCVSTM   DC    7
03B4R 002F             DC    46
03B6R 4348             DC    C'CHANNEL - '
     414E
     4E45
     4C20
     2D20
03C0R 2020             DC    C'    1     2     3  '
     2031
     2020
     2020
     2032
     2020
     2020
     2033
     2020
```

```
03D2R 2020             DC     C'    4      5      6    '
      2034
      2020
      2020
      2035
      2020
      2020
      2036
      2020
03E4R 0007     RCVST2  DC     7
03E6R 002E             DC     46
03E8R 5354             DC     C'STATUS   -  '
      4154
      5553
      2020
      2020
03F2R 2020     RCVSTA  DC     C'                    '
      2020
      2020
      2020
      2020
      2020
      2020
      2020
      2020
0404R 2020             DC     C'                    '
      2020
      2020
      2020
      2020
      2020
      2020
      2020
      2020
               *
0416R 204F     ON      DC     C' ON '
      4E20
041AR 4F46     OFF     DC     C'OFF '
      4620
041ER 2044     DU      DC     C' DU '
      5520
0422R 494C     ILL     DC     C'ILL '
      4C20
               *
0426R          PRTMES  EQU    *
0426R 40C0             STH    R12,CHSTA          STORE CHSTA
      0324R
042AR 40C0             STH    R12,NCHSTA         MAKE NCHASTA = CHSTA
      0330R
042ER 0833     CKCONT  LHR    R3,R3              CHECK IF START TASK MESS.
0430R 2133             BNZS   CKENT              IF NOT, GO TO CKENT
0432R E130     ENDJOB  SVC    3,0                EOJ
      0000
               *
0434R 0834     CKENT   LHR    R3,R4              CHECK IF ENTRY
0436R 2333             BZS    ENDJOB             IF NOT GO TO ENDJOB
```

```
                  *
043AR 6620        CONTIN  RTL    R2,INFO          INPUT ID AND DATA
      004ER
043ER 6640                RTL    R4,INFO          INPUT TIME(HRS & MIN)
      004ER
0442R 6650                RTL    R5,INFO          INPUT TIME(SEC & 120TH'S)
      004ER
0446R 4340                BFC    4,LDDATA         IF LIST ENTRY, GO TO LDDATA
      0452R
044AR E120                SVC    2,ERRLST         PRINT "NO LIST ENTRY"
      02C4R
044ER E130                SVC    3,0              EOJ
      0000
                  *
0452R 0862        LDDATA  LHR    R6,R2            LOAD R6 WITH ID & DATA
0454R C873                LHI    R7,-1(R3)        LOAD R7 WITH CH NO. -1
      FFFF
0458R EB60                RLL    R6,6             COMBINE CH & ID IN R7
      0006
045CR D3B7                LB     R11,ADMTAB(R7)   LOAD R11 WITH SENSOR NUMBER
      0000F
0460R 08BB                LHR    R11,R11          SET FLAGS
0462R 4230                BNZ    LCTNLG           IF NOT 0, GO TO LCTNLG
      0494R
0466R 6110        ERRCHA  AHM    R1,INCTSE        INCR CTR
      0000F
046AR 6110                AHM    R1,INCTSI        INCR CTR
      0000F
046ER E130                SVC    3,0              EOJ
      0000
0472R             INTBUF  DS     ABSZE
                  *
                  *
                  * FIND ADDRESS IN ADMIN LOG
                  *
0494R C8C0        LCTNLG  LHI    R12,@ADMSZ       LOAD R12 WITH RECORD SIZE
      0000F
0498R 27B1                SIS    R11,1            SUBTRACT 1 FROM SENSOR NUMBER
049AR 0CAC                MHR    R10,R12          MULT SENSOR NO. BY SIZE
049CR CAB0                AHI    R11,ACTADM       ADD ADDR OF ADMIN LOG
      0000F
                  *
                  * ROUTINE TO DETERMINE IF DATA IS TO BE OUTPUT
                  * TO THE TEXAS INSTRUMENTS THERMAL RECORDER
                  *
04A0R D39B        RDROUT  LB     R9,@RECNO(R11)   LOAD RECORDER NUMBER
      0000F
04A4R C990                CHI    R9,1             IS DATA TO BE OUTPUT TO RECORDER 1
      0001
04A8R 2133                BNES   UPDATE           IF NOT, CONTINUE ON THRU INP
04AAR 41E0                BAL    LINK1,TEXAS      IF SO, GO TO THERMAL RECORDER ROUT
      0000F
                  *
                  *
                  *
                  *
```

```
                    * UPDATE ADMIN LOG EXTENSION & INCR ACTIV CTR
                    *
04AER 404B          UPDATE  STH   R4,@LTMAC(R11)   STORE TIME (HR & MIN)
      0000F
04B2R 405B                  STH   R5,@LTMA2(R11)   STORE TIME (SEC & 120TH'S)
      0000F
04B6R 611B                  AHM   R1,@NOACT(R11)   INCR ACTIV CTR(SINCE ERASE)
      0000F
04BAR 9466                  EXBR  R6,R6            SHIFT DATA TO RIGHT BYTE
04BCR C460                  NHI   R6,255           REMOVE 1ST BYTE OF R6
      00FF
04C0R D39B                  LB    R9,@SNSTP(R11)   LOAD R9 WITH SLU
      0000F
04C4R 9192                  SLLS  R9,2             MULT BY 4
04C6R D389                  LB    R8,TYPE(R9)      LOAD R8 WITH CLASS
      0000F
04CAR 0981                  CHR   R8,R1            IF CLASS IS NOT 1,SKIP
04CCR 4230                  BNZ   NOCHGE
      04D2R
04D0R 0860                  LHR   R6,R0
04D2R CA88          NOCHGE  AHI   R8,-2(R8)        LOAD R8 WITH 2 X CLASS -2
      FFFE
04D6R 4888                  LH    R8,TBLADR(R8)    LOAD R8 WITH SW ADDR
      0516R
04DAR 0308                  BR    R8               GO TO SW ADDRESS
                    *
04DCR C960          LDCLS   CHI   R6,63            COMPARE DATA WITH 63
      003F
04E0R 2339                  BZS   STACT            IF 63, GO TO STACT
04E2R C960                  CHI   R6,57            COMPARE DATA WITH 57
      0039
04E6R 4270                  BP    BADDAT           IF GTR 57, GO TO BADDAT
      0502R
04EAR C960                  CHI   R6,52            COMP DATA WITH 52
      0034
04EER 4210                  BM    BADDAT           IF LT 52, GO TO BADDAT
      0502R
04F2R D26B          STACT   STB   R6,@LCBR(R11)    STORE DATA
      0000F
04F6R 6110          CNTACT  AHM   R1,ACCTSE        INCR CTR
      0000F
04FAR 6110                  AHM   R1,ACCTSI        INCR CTR
      0000F
04FER 4300                  B     CALCAS
      051AR
                    *
                    *
0502R C880          BADDAT  LHI   R8,X'80'         LOAD R8 WITH DATA BIT
      0080
0506R D28B                  STB   R8,@LCBR(R11)    STORE IN DATA AREA
      04F4R
050AR 6110                  AHM   R1,WDCTSE        INCR CTR
      0000F
050ER 6110                  AHM   R1,WDCTSI        INCR CTR
      0000F
0512R 4300                  B     CNTACT           GO TO CNTACT
```

```
       04F6R
                *
0516R 04F6R     TBLADR  DC    CNTACT
0518R 04DCR             DC    LDCLS
                *
                * CALL SUBROUTINE TO CONVERT SENSOR DATA TO ASCII
                * AND PLACE IN AN INTERIM BUFFER
                *
051AR 40B0      CALCAS  STH   R11,SNSAD1        STORE ADDR OF SENSOR DATA
      0522R
                *
051ER 41E0              BAL   LINK1,CONASC      CALL CONV TO ASCII ROUTINE
      0000F
0522R           SNSAD1  DS    2                 ADDR OF SENSOR DATA
0524R 0472R             DC    INTBUF            ADDR OF INTERIM BUFFER
                *
0526R 4300              B     CRTOUT            BRANCH TO CRTOUT
      0000F
                *
052AR                   END
```

```
                    *
                    *
                    ****************************************
                    *
                    *
                    *   *****  *****   *   *    *     ***
                    *     *    *       *   *   * *   *   *
                    *     *    *        * *   *   *  *
                    *     *    ***       *    *   *   ***
                    *     *    *        * *   *****      *
                    *     *    *       *   *  *   *  *   *
                    *     *    *****   *   *  *   *   ***
                    *
                    *
                    ****************************************
                    *
                    *
0000R                       ENTRY TEXAS,RDRBUF,LASTBF
0000R                       ENTRY TEXRUN
0000R                       EXTRN TYPE,@.TMAC,@.TMA?
0000R                       EXTRN @SNSTF,@FENNO
0000R                       EXTRN RFTA,TSLGRA,FLINK
                    *
                    *
                    * WRITTEN AND DEBUGGED BY RICH MARTINO
                    *
                    *
                    * THIS ROUTINE LOADS SEQUENCIAL BUFFERS WITH DATA
                    * WHICH WILL BE OUTPUT TO A THERMAL RECORDER BY ANOTHER
                    * TASK A SPECIFIED TIME AFTER A DEVICE AVAILABLE SIGNAL
                    * IS RECEIVED FROM THE TI THERMAL RECORDER
                    *
                    * IN THIS VERSION, THE COMPUTER IS THE CHARACTER GENERATOR
                    *
0000R               TEXAS   EQU   *
0000R D000                  STM   R0,TAB-112
      04B0R
0004R 4040                  STH   R4,TAB-36
      0502R
0008R 4050                  STH   R5,TAB-34
      0504R
000CR D3AB                  LB    R10,@FENNO(R11)
      0000F
0010R 40A0                  STH   R10,TAB-32
      0506R
0014R 40B0                  STH   R11,TAB-30
      0508R
0018R 08CA                  LHR   R12,R10
001AR 0200                  NOPR
001CR D39B                  LB    R9,@SNSTF(R11)
      0000F
0020R 9192                  SLLS  R9,2
0022R D389                  LB    R8,TYPE(R9)
      0000F
0026R 4980                  CH    R8,TAB-10
      051CR
```

```
002AR 4330              BFC    3,DETECT
      003AR
002ER 4980              CH     R8,TAB-8
      051ER
0032R 4330              BFC    3,CLASFY
      012ER
0036R 4300              BFC    0,EXIT1
      01D6R
                 *  PROCESSING OF A DETECTION SENSOR
                 *
003AR C88B       DETECT LHI    R8,@LTMAC(R11)
      0000F
003ER 4898              LH     R9,0(R8)
      0000
0042R 4090              STH    R9,TAB-48
      04F6R
0046R 4898              LH     R9,2(R8)
      0002
004AR 4090              STH    R9,TAB-46
      04F8R
004ER 4840              LH     R4,TAB-36
      0502R
0052R 4850              LH     R5,TAB-34
      0504R
0056R 404B              STH    R4,@LTMAC(R11)
      003CR
005AR 405B              STH    R5,@LTMA2(R11)
      0000F
005ER C88B              LHI    R8,@LTMAC(R11)
      0058R
0062R 4898              LH     R9,0(R8)
      0000
0066R 4090              STH    R9,TAB-42
      04FCR
006AR 4898              LH     R9,2(R8)
      0002
006ER 4090              STH    R9,TAB-40
      04FER
                 * ONE MINUTE ACTIVATION COMPARE ROUTINE
                 *
      0711              XHR    R1,R1
      D381              LB     R8,TAB-48(R1)
      04F6R
                        L      R9,TAB-42(R1)

                        SHR    R9,R8
                        CH     R9,TAB-12

                        BFC    3,CKMINS

                        CH     R9,TAB-10

                        BFFS   3,4
                        SRLS   R9,4
                        BFC    0,ZROLOG
```

```
0092R 2611              AIS   R1,1
0094R D381              LB    R8,TAB-48(R1)
      04F6R
0098R D391              LB    R9,TAB-42(R1)
      04FCR
009CR 4980              CH    R8,TAB-14
      0518R
00A0R 2333              BFFS  3,3
00A2R 4300              BFC   0,ZROLOG
      00E0R
00A6R 4990              CH    R9,TAB-12
      051AR
00AAR 4330              BFC   3,CKSECS
      00D0R
00AER 4300              BFC   0,ZROLOG
      00E0R
                * CHECK ON MINUTES ROUTINE
                *
00B2R 2611      CKMINS  AIS   R1,1
00B4R D381              LB    R8,TAB-48(R1)
      04F6R
00B8R D391              LB    R9,TAB-42(R1)
      04FCR
00BCR 0E98              SHR   R9,R8
00BER 4990              CH    R9,TAB-12
      051AR
00C2R 4330              BFC   3,INCLOG
      00E6R
00C6R 4990              CH    R9,TAB-10
      051CR
00CAR 2333              BFFS  3,3
00CCR 230A              BFFS  0,10
00CER 2409              LIS   R0,R9
                * CHECK ON SECONDS ROUTINE
                *
00D0R 2611      CKSECS  AIS   R1,1
00D2R D381              LB    R8,TAB-48(R1)
      04F6R
00D6R D391              LB    R9,TAB-42(R1)
      04FCR
00DAR 0989              CHR   R8,R9
00DCR 2125              BTFS  2,5
00DER 2301              BFFS  0,1
                * ACTIVATION LOG UPDATE ROUTINE
                *
00E0R 0711      ZROLOG  XHR   R1,R1
00E2R D21A              STB   R1,TABLE+650(R10)
      0466R
00E6R D31A      INCLOG  LB    R1,TABLE+650(R10)
      0466R
00EAR 4910              CH    R1,TAB-2
      0524R
00EER 2122              BTFS  2,2
00F0R 2611              AIS   R1,1
00F2R D21A              STB   R1,TABLE+650(R10)
      0466R
```

```
00F6R 4910              CH     R1,TAB-10
      051CR
00FAR 2135              BTFS   3,5
00FCR D380              LB     R8,TABLE+71
      0223R
0100R 4300              BFC    0,STORE
      0120R
0104R 4910              CH     R1,TAB-8
      051ER
0108R 2134              BTFS   3,4
010AR D380              LB     R8,TABLE+73
      0225R
010ER 2309              BFFS   0,9
0110R 4910              CH     R1,TAB-6
      0520R
0114R 2134              BTFS   3,4
0116R D380              LB     R8,TABLE+75
      0227R
011AR 2303              BFFS   0,3
011CR D380              LB     R8,TABLE+77
      0229R
0120R 4890       STORE  LH     R9,RPTR
      0000F
0124R 0A9C              AHR    R9,R12
0126R D289              STB    R8,0(R9)
      0000
012AR 4300              BFC    0,EXIT1
      01D6R
                 * PROCESSING OF A CLASSIFICATION SENSOR
                 *
012ER 9466       CLASFY EXBR   R6,R6
0130R C460              NHI    R6,255
      00FF
0134R 4960              CH     R6,TAB-26
      050CR
0138R 2135              BTFS   3,5
013AR C880              LHI    R8,TABLE+14
      01EAR
013ER 2307              BFFS   0,7
0140R 2036              BTBS   3,6
0142R 4960              CH     R6,TAB-24
      050ER
0146R 2135              BTFS   3,5
0148R C880              LHI    R8,TABLE+42
      0206R
014CR 2307              BFFS   0,7
014ER 2254              BFBS   5,4
0150R 4960              CH     R6,TAB-22
      0510R
0154R 2135              BTFS   3,5
0156R C880              LHI    R8,TABLE+0
      01DCR
015AR 2307              BFFS   0,7
015CR 2037              BTBS   3,7
015ER 4960              CH     R6,TAB-20
      0512R
```

```
0162R 2134              BTFS   3,4
0164R C880              LHI    R8,TABLE+28
      01F8R
0168R 2309              BFFS   0,9
016AR 4960              CH     R6,TAB-18
      0514R
016ER 2134              BTFS   3,4
0170R C880              LHI    R8,TABLE+28
      01F8R
0174R 2303              BFFS   0,3
0176R C880              LHI    R8,TABLE+56
      0214R
                  * CHARACTER / BUFFER  TRANSPOSITION ROUTINE
                  *
017AR 24D5              LIS    R13,5
017CR 40D0              STH    R13,TSK6PR
      0000F
0180R 40D0              STH    R13,TAB-28
      050AR
0184R D000              STM    R0,TAB-80
      04D6R
0188R 4110              BAL    R1,FLINK
      0000F
018CR D100              LM     R0,TAB-80
      04D6R
0190R 48E0              LH     R14,RPTR
      0122R
0194R 24D7              LIS    R13,7
0196R 268D              AIS    R8,13
0198R 0AEC              AHR    R14,R12
                  * DATA TRANSFER ROUTINE
019AR D378        NXLINE LB     R7,0(R8)
      0000
019ER D27E              STB    R7,0(R14)
      0000
01A2R C9E0              CHI    R14,TABLE+570
      0416R
01A6R 2339              BFFS   3,9
01A8R 2128              BTFS   2,8
01AAR CAE0              AHI    R14,80
      0050
01AER 2782              SIS    R8,2
01B0R 27D1              SIS    R13,1
01B2R 4230              BTC    3,NXLINE
      019AR
01B6R 2308              BFFS   0,8
01B8R C8E0              LHI    R14,TABLE+90
      0236R
01BCR 0AEC              AHR    R14,R12
01BER 2782              SIS    R8,2
01C0R 27D1              SIS    R13,1
01C2R 4230              BTC    3,NXLINE
      019AR
01C6R 0700              XHR    R0,R0
01C8R 4000              STH    R0,TAB-28
      050AR
```

```
01CCR 24D3              LIS    R13,3
01CER 40D0              STH    R13,TSK6FR
      017ER
01D2R 4110              BAL    R1,FLINK
      018AR
            * NORMAL EXIT ROUTINE
            *
01D6R D100  EXIT1       LM     R0,TABLE+730
      04B6R
01DAR 030E              BR     LINK1
            *
            * TABULAR INDEX OF DEFINED
            * CONSTANTS AND STORAGE AREA
            *
01DCR       TABLE       EQU    *
01DCR 001F              DC     31,04,04,04,04,04,04,30,17
      0004
      0004
      0004
      0004
      0004
      0004
      001E
      0011
01EER 0011              DC     17,30,16,16,16,04,10,17,17
      001E
      0010
      0010
      0010
      0004
      000A
      0011
      0011
0200R 001F              DC     31,17,17,17,17,17,21,21,21
      0011
      0011
      0011
      0011
      0011
      0015
      0015
      0015
0212R 000A              DC     10,04,21,14,04,14,21,04,04
      0004
      0015
      000E
      0004
      000E
      0015
      0004
      0004
0224R 000A              DC     10,21,31,00,00,00,00,00,00
      0015
      001F
      0000
      0000
```

```
      0000
      0000
      0000
      0000
0236R                DS    726
050CR 0035           DC    53,54,55,56,57,58,X'59'
      0036
      0037
      0038
      0039
      003A
      0059
051AR 0000           DC    00,01,02,03,04,05,06
      0001
      0002
      0003
      0004
      0005
      0006
0000          R0     EQU   0
0001          R1     EQU   1
0002          R2     EQU   2
0003          R3     EQU   3
0004          R4     EQU   4
0005          R5     EQU   5
0006          R6     EQU   6
0007          R7     EQU   7
0008          R8     EQU   8
0009          R9     EQU   9
000A          R10    EQU   10
000B          R11    EQU   11
000C          R12    EQU   12
000D          R13    EQU   13
000E          R14    EQU   14
000F          R15    EQU   15
000E          LINK1  EQU   14
050AR         TEXRUN EQU   TABLE+814
0416R         LASTBF EQU   TABLE+570
0236R         RDRBUF EQU   TABLE+90
0526R         TAB    EQU   *-2
0528R                END
```

TASK 4
OPERATOR COMMAND TASK

```
 - OPERATOR COMMAND PROGRAMS FOR THE SENSOR MONITOR SET     PAGE    1
                    *
                    *  OPERATOR CONTROL COMMANDS  OVERLAY AREA
                    *
                    *  REWRITTEN  AND  DEBUGGED  BY :
                    *  RICH MARTINO  &  JOE KARAKOWSKI
                    *
                    *
                    *
                    *
                    *  OPERATOR COMMANDS  TASK CONTROL BLOCK
                    *
0000R                         ENTRY OVERLY,OVERLE
0000R                         ENTRY TASK4,CMD
0000R               TASK4     DS    8                *UNUSED
0008R               ADRS      DS    2                *ADDRESS INPUT SLOT
000AR                         DS    2                *UNUSED
000CR                         DS    2                *UNUSED TELL BFR
000ER 0406                    DC    X'0406'          *LU 00--DISC HISTORY FILE
0010R 0010                    DC    X'10'            *LU 01--ALERT CRT
0012R 0000                    DC    0                LU 2--UNUSED
0014R 0062                    DC    X'62'            *LU 03--LINE PRINTER
0016R 0106                    DC    X'106'           *LU 04--DISC ADMIN FILE
0018R 0206                    DC    X'206'           *LU 05--DISC OVERLAY FILE
001AR 0013                    DC    X'13'            *LU 6 HSPTR
001CR 0306                    DC    X'306'           *LU 07--TPL HISTORY FILE
001ER 0006                    DC    X'C6'            *LU 08--RESIDENT SOFTWARE
0020R 0000                    DC    0,0              *UNUSED LUS
      0000
0024R 0000                    DC    0,0,0,0,0
      0000
      0000
      0000
      0000
002ER                         DS    32               * GR SAVE
                    *
                    *
                    *TASK 4 PROGRAM
                    *
                    *
004ER               CMD       EQU   *
000E                R14       EQU   14
000F                R15       EQU   15
004ER 48E0                    LH    R14,ADRS
      0008R
0052R 48EE                    LH    R14,0(R14)       LOAD BUFF ADRS INTO R14
      0000
                    * R14---TTY BUFFER
                    * R15---RETURN ADDRESS REGISTER
0056R 41F0                    BAL   R15,OVERLY
      005ER
005AR E130                    SVC   3,0
      0000
005ER               OVERLY    DS    X'1200'
125DR               OVERLE    EQU   *-1
125ER                         END
```

TASK 5
COMMANDS DIRECTORY TASK

```
 - COMMANDS DIRECTORY FOR THE SENSOR MONITOR SET              PAGE    1
                 *
                 *
                 *    COMMANDS DIRECTORY ROUTINE
                 *
                 *
                 *     REWRITTEN  AND  DEBUGGED  BY  :
                 *      RICH MARTINO  &  JOE KARAKOWSKI
                 *
                 *
                 *
                 *
0000R                     EXTRN ALGFLG
0000R                     EXTRN OVERLY,OVERLE
0000R                     EXTRN T4STAT
0000R                     ENTRY TIMCOM
0000R                     ENTRY SRU,SRUTB
                 *
                 *   COMMANDS DIRECTORY  TASK CONTROL BLOCK
                 *
0000R            SRUTB    DS    8               UNUSED
0008R 0000                DC    0               PARAMETER
000AR                     DS    2               UNUSED
000CR 03BER               DC    TTYBUF          TELL BUFFER
000ER 0000                DC    0               LU 0
0010R 0002                DC    2               LU 1--TTY
0012R 0000                DC    0
0014R 0062                DC    X'62'           LU 3--LINE PRINTER
0016R 0062                DC    X'62'           LU 4 = LINE PRINTER
0018R 0013                DC    X'13'           LU 5--TAPE
001AR 0206                DC    X'206'          LU 6--DISC OVERLAY FILE
001CR 0002                DC    2               LU 7--TTY
001ER 0000                DC    0,0,0,0         LU 8-11
      0000
      0000
      0000
0026R 0000                DC    0,0,0,0         LU 12-15
      0000
      0000
      0000
002ER                     DS    32              GEN REG SAVE
```

```
                    *
                    *
                    *
                    *
                    *   COMMANDS DIRECTORY  MAIN PROGRAM
                    *
                    *
                    *
004ER               SRU     EQU    *
004ER E110                  SVC    1,DIRRD          READ DIRECTORY INTO BUFFER
      0152R
0052R 4800                  LH     STAT,DIRRD+2     TEST STATUS
      0154R
0056R 4210                  BM     DEVER1
      00F8R
005AR 24E1                  LIS    ONE,1
005CR 07CC                  XHR    ZRO,ZRO          ZERO REG 0
005ER 40C0                  STH    ZRO,ALGFLG       ZERO ALGOR FLAG
      0000F
0062R 40C0                  STH    ZRO,TTYBUF
      03BER
0066R               TASK5   EQU    *
0066R 4820                  LH     TS2,T4STAT       REQUEST STATUS OF TK4
      0000F
006AR C520                  CLHI   TS2,X'8000'      TEST IF DORMANT
      8000
006ER 4230                  BNE    ALGOR
      00FCR
0072R 4810                  LH     TS1,TTYBUF       IS THERE A COMMAND TO PROCESS
      03BER
0076R 4330                  BZ     ALGOR            IF ZERO CHECK FOR ALGOR
      00FCR
007AR 41D0          PRMSER  BAL    RTN,SEARCH
      00CAR
007ER 40C0                  STH    ZRO,TTYBUF       ZERO TTY BUFFER
      03BER
0082R 4300                  B      ALGOR            GO TO CHECK ALGORITHM
      00FCR
0086R 4520                  CLH    TS2,PROGRD+8     ALREAY IN CORE ?
      03B8R
008AR 4330                  BE     CALTK4           YES,SKIP  READ
      00A2R
008ER 4020                  STH    TS2,PROGRD+8     SET UP  PARAMETER
      03B8R
0092R 4020                  STH    TS2,TASK4+14     BLOCKS
      037CR
0096R E110                  SVC    1,PROGRD         READ  IN PROG
      03B0R
009AR 4800                  LH     STAT,PROGRD+2    TEST STATUS
      03B2R
009ER 4210                  BM     DEVER2
      014AR
00A2R E160          CALTK4  SVC    6,TASK4
      03FFR
00A6R 4330                  BZ     TASK5
      0066R
```

```
00AAR E120     TKERR    SVC    2,TASKER
      00B2R
00AER 4300              B      TASK5
      0066R
00B2R 0007     TASKER   DC     7,20,C' THERE IS A TASK ERR'
      0014
      2054
      4845
      5245
      2049
      5320
      4120
      5441
      534B
      2045
      5252
00CAR          SEARCH   EQU    *
00CAR 4840              LH     CT,DIRBUF+2        LOAD NO. OF ENTR AS COUNT
      015ER
00CER 9142              SLIS   CT,2               SL(MULTX2)TO MAKE COUNT AN ADR
00D0R 0733              XHR    DIPO,DIPO          RESET DIR POINTER
00D2R 4513     NAMSER   CLH    TS1,DIRBUF+4(DIPO) CHECK NAME WITH DIRECTORY
      0160R
00D6R 4330              BE     FILFBK             IF  NAME FOUND THEN BRANCH
      00F0R
00DAR 2634              AIS    DIPO,4             INCR DIR PNTR
00DCR 0534              CLHR   DIPO,CT            TEST IF  SEARCH  IS FINISH
00DER 4280              BL     NAMSER             OTHERWISE  CONT
      00D2R
00E2R E120              SVC    2,CMERR            ERROR NAME  NOT  FOUND
      038ER
00E6R 4010              STH    TS1,JOE            PUT BAD CMD IN BUFFER
      03AER
00EAR E120              SVC    2,ERMS             WRITE OUT CMD
      03AAR
00EER 030D              BR     RTN                RETURN TO ROUTINE
00F0R          FILFBK   EQU    *
00F0R 4823              LH     TS2,DIRBUF+6(DIPO) LOAD SECTOR ADR IN TS2
      0162R
00F4R 430D              B      8(RTN)             RETURN TO CALLER
      0008
00F8R 4300     DEVER1   B      DEVER2
      014AR
00FCR          ALGOR    EQU    *
00FCR 45E0              CLH    ONE,ALGFLG
      0060R
0100R 4230              BNE    TASK5
      0066R
0104R E160              SVC    6,TASK6
      037ER
0108R 4230              BNZ    TKERR
      00AAR
010CR 4000              STH    ZRO,ALGFLG         ZERO ALGORITHM FLAG
      00FER
0110R 4300              B      TASK5
      0066R
```

```
                *
                *
                *
                *      TIME COMPACT ROUTINE
                *     THIS ROUTINE  CALLS THE TIME OF DAY HH:MM:SS
                *     AND COMPRESSES THE : I.E. HHMMSS
                *    ONE MUST LOAD GEN REG 1(TS1) WITH THE ADR WHERE THE RESULT I
                *     STORED
                *
                *
0114R           TIMCOM  EQU    *
0114R E120              SVC    2,RDTIME          READ CURRENT TIME
      013CR
0118R 48A0              LH     R10,RDTIME+10
      0146R
011CR 40A1              STH    R10,4(TS1)        STORE SS IN TN
      0004
0120R 48A0              LH     R10,RDTIME+4
      0140R
0124R 40A1              STH    R10,0(TS1)        STORE HH IN TN+4
      0000
0128R 48A0              LH     R10,RDTIME+6      /:M/
      0142R
012CR 91A8              SLLS   R10,8             SHIFT SO/MO/
012ER 48B0              LH     R11,RDTIME+8      /M:/
      0144R
0132R 90B8              SRLS   R11,8             SHIFT SO /OM/
0134R 06AB              OHR    R10,R11           OR /MO/+/OM/
0136R 40A1              STH    R10,2(TS1)        STORE MM IN TN+2
      0002
013AR 030D              BR     RTN               RETURN
                *
                *
                *
                *
                *
                *       D  C   CONSTANTS
                *
                *
                *
013CR 0008      RDTIME  DC     8,*+2,0,0,0,0
      0140R
      0000
      0000
      0000
      0000
0148R 0000      LSTMIN  DC     0
014AR           DEVER2  EQU    *
014AR E120              SVC    2,ERMESS
      035ER
014ER 4300              B      TASKS
      0066R
0152R 5C06      DIRRD   DC     X'5C06',0,DIRBUF,DIREND,0
      0000
      015CR
      035BR
```

```
        0000
015CR             DIRBUF   DS     512
035BR             DIREND   EQU    *-1
0000              STAT     EQU    0
0001              TS1      EQU    1
0002              TS2      EQU    2
0003              R3       EQU    3
0003              DIPD     EQU    3
0004              CT       EQU    4
0005              R5       EQU    5
0006              R6       EQU    6
0007              R7       EQU    7
0008              R8       EQU    8
0009              ALCT     EQU    9
000A              R10      EQU    10
000B              CMD      EQU    11
000B              R11      EQU    11
000C              ZRO      EQU    12
000D              RTN      EQU    13
000F              ONE      EQU    14
000F              R15      EQU    15
035CR 2020        BLANK    DC     X'2020'
035ER 0007        ERMESS   DC     7,10,C'I/O ERROR '
      000A
      492F
      4F20
      4552
      524F
      5220
036CR 0001        PAUSE    DC     1
036ER 434D        TASK4    DC     C'CMD    ',X'0202',0,*+2,TTYBUF,0
      4420
      2020
      0202
      0000
      037AR
      03BER
      0000
037ER 5441        TASK6    DC     C'TASK6 ',X'202',0,0,0,0
      534B
      3620
      0202
      0000
      0000
      0000
      0000
038FR 0007        CMERR    DC     7,24,C' CMD-ERR ,INVALID CMD IS'
      0018
      2043
      4D44
      2D45
      5252
      202C
      494E
      5641
      4C49
```

```
        4420
        434D
        4420
        4953
03AAR 0007     ERMS     DC     7,2
        0002
03AER 0000     JOE      DC     0
03B0R 5006     PROGRD   DC     X'5006',0,OVERLY,OVERLE,0
        0000
        0000F
        0000F
        0000
               *    TTYBUF IS FROM TASK1
               * THE FOLLOWING CARD MUST BE IMMEDIATELY BEFORE THE TTYBUF
               *
               ****
               ***
               **
               *
03BAR 0007              DC     7,80
        0050
03BER          TTYBUF   DS     80                   TELL BUFFER
040ER          RELORG   EQU    *
0001           CRT1LU   EQU    X'0001'
0002           CRTLU2   EQU    X'0002'
0002           CRT2LU   EQU    X'0002'
0005           OVLFIL   EQU    X'0005'
2000           WRITE    EQU    X'2000'
0800           WAIT     EQU    X'0800'
0004           LNFLU    EQU    X'0004'
0400           RANDOM   EQU    X'0400'
0000           ASCII    EQU    X'0000'
040ER 0007     ABSERR   DC     7,27,C'PROGRAM IS NOT IN DIRECTORY'
        001B
        5052
        4F47
        5241
        4D20
        4953
        204E
        4F54
        2049
        4E20
        4449
        5245
        4354
        4F52
        5920
042ER                   END    SRU
```

TASK 6

TACTICAL SITUATION SIMULATOR TASK

```
TACTICAL SITUATION SIMULATION PROGRAM                                 PAGE    1
                  * AUTHOR    J. KARAKOWSKI
                  *
0000R                     ENTRY @SEGNO
0000R                     ENTRY @OBJCL,@SENP,@SENW,@SENTK,@SENUK
0000R                     ENTRY TASK6,TSK6TB,CNVTAB,SENSNO,ACTSEN
0000R                     ENTRY CURSEG,CURSEN,CUROBJ,@SEGLT
0000R                     ENTRY @SENEN,OBJLST,@STRNG,STRTBL
0000R                     ENTRY SEGLST,SEGNUM,@SENRW
0000R                     ENTRY STRNO,@SEG1X,@SEG1Y,@SEG2X,@SEG2Y
0000R                     ENTRY @SEGL,@SEGM,STRLGT,OBJNUM
0000R                     ENTRY @LSTL,@OBJNO,@OBJST,@OBJD,@OBJVE,@OBJRS,@OBJEN
0000R                     ENTRY @SEGPO,@OBJFG,@SENID,@SENFG,@SENRT,@SENTP
0000R                     ENTRY @SENTB,@SENSX,@SENSY,@SENPB,@SENRR
0000R                     EXTRN .U,SORT, W
0000R                     EXTRN INFO,ALGFLG
0000R                     EXTRN CRT1DA,TSK3OP
0000R 0398R      TSK6TB   DC     TASK6,TASK6
      0398R
0004R 006ER               DC     UTOP
0006R 0000R               DC     TSK6TB
0008R 0000                DC     0                    PARAMETER
000AR                     DS     2                    UNUSED
000CR                     DS     2                    TELL BUFF
000ER 0000                DC     0                    LU 1
0010R 0000                DC     0                    LU 1
0012R 0062                DC     X'62'                LU 2
0014R 0406                DC     X'406'               LU 3
0016R 0010                DC     X'10'                LU 4
0018R 0002       LU5      DC     2                    LU 5
001AR                     DO     10
001AR 0000                DC     0
001CR 0000                DC     0
001ER 0000                DC     0
0020R 0000                DC     0
0022R 0000                DC     0
0024R 0000                DC     0
0026R 0000                DC     0
0028R 0000                DC     0
002AR 0000                DC     0
002CR 0000                DC     0
002ER                     DS     64                   GEN+FLT PT REGSAV
006ER            UTOP     DS     700
0000             R0       EQU    0
0001             R1       EQU    1
0002             R2       EQU    2
0003             R3       EQU    3
0004             R4       EQU    4
0005             R5       EQU    5
0006             R6       EQU    6
0007             R7       EQU    7
0008             R8       EQU    8
0009             R9       EQU    9
000A             R10      EQU    10
000B             R11      EQU    11
000C             R12      EQU    12
000D             R13      EQU    13
```

AD-A103 991 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND FO--ETC F/G 9/2
TACTICAL SITUATION SIMULATOR ALGORITHM FOR USE WITH A THERMAL L--ETC(U)
AUG 81 J A KARAKOWSKI, R J MARTINO, A SLUTSKY
UNCLASSIFIED DELCS-TR-81-1 NL

2 OF 2
AD A
103991

END
DATE
FILMED
10-81
DTIC

```
000E             R14     EQU    14
000F             R15     EQU    15
0004             ALRTLU  EQU    4
0000             SCX     EQU    0
0004             RX      EQU    4
0006             INCX    EQU    6
0008             XR      EQU    8
000A             SCY     EQU    10
000E             RY      EQU    14
0010             INCY    EQU    16
0012             YR      EQU    18
032AR            SCXF    DS     4
032ER            RXF     DS     4
0332R            INCXF   DS     4
0336R            XRF     DS     4
033AR            SCYF    DS     4
033ER            RYF     DS     4
0342R            INCYF   DS     4
0346R            YRF     DS     4
034AR            XLOC    DS     2
034CR            YLOC    DS     2
034ER            MAPFLG  DS     2
0350R            MAPWT   DS     2
0352R 2004       OBJLOC  DC     X'2000'+ALRTLU
0354R                    DS     2
0356R 035CR              DC     MAPBUF,MAPBUF+3
      035FR
035AR                    DS     2                   UNUSED
035CR 6C00       MAPBUF  DC     X'6C00'
035ER 002A               DC     X'2A'
0360R                    DS     4                   UNUSED
0364R 4232       MAXM    DC     X'4232',0
      0000
0368R C232       MINM    DC     X'C232',0
      0000
036CR            OLDX    DS     2
036ER            OLDY    DS     2
0370R 0008       C8      DC     8
0372R 4048       COP5    DC     X'4048',0
      0000
0376R 4099       COP6    DC     X'4099',X'999A'
      999A
037AR 4CCC       CLASSG  DC     X'4CCC',X'CCCD'
      CCCD
037ER 4040       TOFFST  DC     X'4040',0
      0000
0382R            II      DS     2
0384R            CLASFG  DS     2
0386R 6666       GATE    DC     X'6666'
0388R 0383       C899    DC     X'383'
038AR 4232       C50     DC     X'4232',0
      0000
038ER 4110       C1      DC     X'4110'
0390R 0000       ZERO    DC     0,0
      0000
0394R 4CCC       FRGATE  DC     X'4CCC',X'CCCD'
```

```
          CCCD
02EA                CLKCNT  EQU   X'2EA'
02DE                SEC     EQU   X'2DE'
003C                CLKINT  EQU   60                 MINIMUM SCEN INTERVAL=0.5 SEC
0398R               TASK6   EQU   *
0398R C8B0                  LHI   R11,UTOP
      006ER
039CR 40B0                  STH   R11,TSK6TB+4
      0004R
03A0R C8B0                  LHI   R11,TSK6TB
      0000R
03A4R 40B0                  STH   R11,TSK6TB+6
      0006R
03A8R 41F0                  BAL   15,.U
      0000F
03ACR D0C0                  STM   R12,FORSAV
      161CR
                    *  ZERO ALGORITHM FLAG
03B0R 0700                  XHR   R0,R0
03B2R 2411                  LIS   R1,1
03B4R 4000                  STH   R0,ALGFLG
      0000F
03B8R 4000                  STH   R0,MAPFLG
      034ER
03BCR 2B00                  SER   R0,R0
03BER 6000                  STE   R0,SCETM
      251ER
03C2R 4000                  STH   R0,SCETIM
      24FCR
03C6R C820                  LHI   R2,X'4080'
      4080
03CAR 4020                  STH   R2,DELTM
      251AR
03CER 4000                  STH   R0,DELTM+2
      251CR
                    *  ZERO MAP WAIT FLAG
03D2R 07AA                  XHR   R10,R10
03D4R 24BA                  LIS   R11,10
03D6R 40B0                  STH   R11,MAPWT
      0350R
03DAR C8B0                  LHI   R11,CRT1DA
      0000F
03DER C8AB                  LHI   R10,2(R11)
      0002
03E2R 40A0                  STH   R10,OUT1
      03ECR
                    *  SET UP CRT PARAMETERS
03E6R 41E0                  BAL   R14,CNVFP
      2216R
03EAR 03DCR                 DC    CRT1DA
03ECR               OUT1    DS    2
03EER 6080                  STE   R8,SCXF
      032AR
03F2R C8AB                  LHI   R10,10(R11)
      000A
03F6R 40A0                  STH   R10,OUT5
```

```
        0404R
03FAR 26A2              AIS    R10,2
03FCR 40A0              STH    R10,OUTS+2
        0406R
                *  SET UP CRT PARAMETERS
0400R 41E0              BAL    R14,CNVFP
        2216R
0404R           OUTS    DS     4
0408R 6080              STE    R8,SCYF
        033AR
040CR 4000              STH    R0,OLDX
        036CR
0410R 4000              STH    R0,OLDY
        036ER
0414R C8B0              LHI    R11,X'435'
        0435
0418R 07AA              XHR    R10,R10
041AR 4CA0              MH     R10,C899
        0388R
041ER C4B0              NHI    R11,X'7FFF'
        7FFF
0422R 40B0              STH    R11,II
        0382R
                *  SYSTEM TIMER LOOP
0426R 4820      WAITCK  LH     R2,CLKCNT
        02EA
042AR 2338              BFFS   3,8
042CR 4200              NOP
        0000
                *  WAIT FOR SCENARIO INTERVAL
0430R E120              SVC    2,WAITI
        0534R
0434R 2207              BFBS   0,7
0436R 4700              NOP
        0000
043AR           CLKSET  EQU    *
043AR 4020              STH    R2,CLK1
        1E84R
043ER 4020              STH    R2,DELTIM
        24FAR
0442R 41D0              BAL    R13,CNVSEC
        050ER
0446R 4060              STH    R6,SEC1
        1E86R
044AR 4000      NXTIME  STH    R0,RESULT       ZERO RESULT ACCUM
        0532R
044ER 4850              LH     R5,CLKCNT       LOAD CURR INTERRUPT COUNT
        02EA
0452R 41D0              BAL    R13,CNVSEC
        050ER
0456R 4560              CLH    R6,SEC1         CCOMPARE CURR WITH PREV TIME
        1E86R
045AR 2333              BFFS   3,3
045CR 41D0              BAL    R13,SECADJ
        049CR
0460R 4550      CLKCHK  CLH    R5,CLK1
```

```
       1E84R
0464R 2323                BFFS  2,3
0466R CB50                SHI   R5,120
       0078
046AR 4870                LH    R7,CLK1
       1E84R
046ER 0B75                SHR   R7,R5
0470R 4A70                AH    R7,RESULT
       0532R
             *  CHECK CLOCK INTERVAL
0474R C570                CLHI  R7,CLKINT
       003C
0478R 4380                BFC   8,STTSK6
       04C0R
047CR C850                LHI   R5,CLKINT
       003C
0480R CB57                SHI   R5,1(R7)
       0001
0484R 4330                BFC   3,NXTIME
       044AR
0488R 0722                XHR   R2,R2
048AR 0835                LHR   R3,R5
048CR 4C20                MH    R2,C8
       0370R
0490R 4030                STH   R3,CLKWT
       052ER
             *  WAIT X MILLISECONDS
0494R E120                SVC   2,WAITM
       052AR
0498R 4300                BFC   0,NXTIME
       044AR
049CR        SECADJ       EQU   *
049CR 2385                BFFS  8,5
049ER 4200                NOP
       0000
04A2R CA60                AHI   R6,60
       003C
04A6R 4B60                SH    R6,SEC1
       1E86R
             *  IF DIFFERENCE=1?
04AAR C560                CLHI  R6,1
       0001
04AER 033D                BFCR  3,R13
04B0R C876                LHI   R7,-1(R6)
       FFFF
04B4R 0766                XHR   R6,R6
04B6R 4C60                MH    R6,ONE20
       0576R
             *  ADJUST SECOND COUNT
04BAR 4070                STH   R7,RESULT
       0532R
04BER 030D                BFCR  0,R13
04C0R        STTSK6       EQU   *
04C0R 6170                AHM   R7,SCETIM
       24FCR
04C4R 4070                STH   R7,DELTIM
```

```
       24FAR
04C8R D1C0              LM     R12,FORSAV
       161CR
04CCR 48B0              LH     11,DELTIM
       24FAR
04D0R 6880              LE     8,$C2
       2556R
04D4R 08EB              LHR    R14,R11
04D6R 41F0              BAL    15,.W
       0000F
04DAR 2800              LER    0,0
04DCR 2D08              DER    0,8
04DER 6000              STE    0,DELTM
       251AR
04E2R D0C0              STM    R12,FORSAV
       161CR
04E6R 0700              XHR    R0,R0
04E8R 2411              LIS    R1,1
04EAR 6800              LE     R0,SCETM
       251ER
04EER 6A00              AE     R0,DELTM
       251AR
                  *  SAVE SCENARIO TIME
04F2R 6000              STE    R0,SCETM
       251ER
                  *  CONVERT SECONDS
04F6R 41D0              BAL    R13,CNVSEC
       050ER
04FAR 4060              STH    R6,SEC1
       1E86R
04FER 4850              LH     R5,CLKCNT
       02EA
0502R 4050              STH    R5,CLK1
       1E84R
                  *  SET CURRENT TIME
0506R E120              SVC    2,RDTIME
       2478R
050AR 4300              BFC    0,STTASK
       0578R
050ER             CNVSEC  EQU    *
                  * USES R6-R9
                  * CONVERTS ASCII SECONDS TO BINARY
050ER 0788              XHR    R8,R8
0510R 4860              LH     R6,SEC
       020E
0514R 0896              LHR    R9,R6
0516R C460              NHI    R6,X'F'
       000F
051AR C490              NHI    R9,X'F00'
       0F00
051ER 9499              EXBR   R9,R9
0520R 4C80              MH     R8,TEN
       0530R
                  *  CONVERT SECONDS
0524R 0A69              AHR    R6,R9
0526R 030D              BFCR   0,R13
```

```
0528R           SYNC     DS     2
052AR 000B      WAITM    DC     11,0
      0000
052ER 0000      CLKWT    DC     0
0530R 000A      TEN      DC     10
0532R           RESULT   DS     2
0534R 000B      WAITI    DC     11,0,1
      0000
      0001
053AR 000B      WAITIX   DC     11,0,9
      0000
      0009
0540R 000B      WAITC    DC     11,0,100
      0000
      0064
0546R 000C      XII      DC     12
0548R 494E      STASK3   DC     C'INP    ',X'2020'
      5020
      2020
      0202
0550R 0000      STSTAT   DC     0,0,0,0
      0000
      0000
      0000
0558R 0007      TASKER   DC     7,20
      0014
055CR 5441               DC     C'TASK3 START ERROR '
      534B
      3320
      5354
      4152
      5420
      4552
      524F
      5220
056ER           ERRNO    DS     2
0570R 0019      XXV      DC     25
0572R 0003      THREE    DC     3
0574R 0000      ACTFLG   DC     0
0576R 0078      ONE20    DC     120
0578R           STTASK   EQU    *
0578R 41D0               BAL    R13,SCENAR        START SCENARIO
      1E8CR
                * OUTPUT ACTIVATIONS FOR PREVIOUS PERIOD
057CR 4000               STH    R0,ACTFLG
      0574R
                *  SET UP LOOP
0580R C830               LHI    R3,ACTSEN
      1634R
0584R 0744               XHR    R4,R4
0586R 4850               LH     R5,SENSNO
      1E8AR
058AR 2751               SIS    R5,1
058CR C860               LHI    R6,X'22'
      0022
0590R 0C46               MHR    R4,R6
```

```
0592R 0846          LHR    R4,R6
0594R 0A53          AHR    R5,R3
               *  R4= ZERO
0596R 6840          LE     R4,ZERO
      0390R
               *  NEXT ACTIVATION
059AR          NXTACT  EQU    *
059AR 6863          LE     R6,8(R3)
      0008
059ER 4320          BFC    2,ZR01
      05C0R
05A2R 6B60          SE     R6,DELTM
      251AR
05A6R 4320          BFC    2,ZR01
      05C0R
05AAR 6960          CE     R6,COPS
      0372R
05AER 2125          BTFS   2,5
05B0R 4200          NOP
      0000
               *  ZERO ANTICIPATE FLAG
05B4R D203          STB    R0,2(R3)
      0002
05B8R 6063          STE    R6,8(R3)
      0008
05BCR 4300          BFC    0,INCR1
      05DER
05C0R          ZR01    EQU    *
05C0R 6043          STE    R4,8(R3)
      0008
               * ZERO SENFLG
05C4R D203          STB    R0,2(R3)
      0002
05C8R 6803          LE     R0,X'14'(R3)
      0014
05CCR 6900          CE     R0,PRGATE        CHECK FOR PROB THRESHOLD
      0394R
05D0R 2187          BTFS   8,7
05D2R 4200          NOP
      0000
05D6R 41D0          BAL    R13,AMDLST
      23AAR
05DAR 4010          STH    R1,ACTFLG
      0574R
               *  ZERO PROBABILITY ACCUMULATOR
05DER          INCR1   EQU    *
05DER 6043          STE    R4,X'14'(R3)
      0014
05E2R D203          STB    R0,X'1C'(R3)
      001C
05E6R D203          STB    R0,X'1D'(R3)
      001D
05EAR D203          STB    R0,X'1E'(R3)
      001E
               *  RESTORE CLASSIFICATION TO ZERO
05EER D203          STB    R0,X'1F'(R3)
```

```
      001F
05F2R C130          BXLE   R3,NXTACT
      059AR
05F6R 4880          LH     R8,MAPFLG
      034ER
05FAR 4580          CLH    R8,MAPWT
      0350R
05FER 2186          BTFS   8,6
0600R 4200          NOP
      0000
0604R 4000          STH    R0,MAPFLG
      034ER
0608R 2303          BFFS   0,3
            *  MAP CHECK
060AR 6110          AHM    R1,MAPFLG
      034ER
060ER 4890          LH     R9,ACTFLG
      0574R
0612R 4330          BFC    3,WAIT1
      063CR
0616R 4000          STH    R0,ACTFLG
      0574R
061AR C890          LHI    R9,X'8000'
      8000
            *  SYSTEM TASK3 SYNC
061ER 4090          STH    R9,TSK3OP
      000F
0622R E160          SVC    6,STASK3
      0548R
0626R 4890          LH     R9,STSTAT
      0550R
062AR 2339          BFFS   3,9
062CR 4200          NOP
      0000
0630R C690          OHI    R9,X'3030'
      3030
0634R 4090          STH    R9,ERRNO
      056ER
0638R E120          SVC    2,TASKER
      0558R
063CR E120  WAIT1   SVC    2,WAITC
      0540R
0640R 4300          BFC    0,NXTIME
      044AR
0644R       SETUP   EQU    *
0644R D060          STM    R6,RSAV1
      0708R
            *  LOAD CURR OBJECT STRING
0648R D375          LB     R7,1(R5)
      0001
064CR 0766          XHR    R6,R6
064ER 2771          SIS    R7,1
0650R C880          LHI    R8,STRLGT
      0011
0654R 0C68          MHR    R6,R8
0656R CA70          AHI    R7,STRTBL          DEVELOP STRING TABLE INDE(
```

```
        1CD8R
                    *  LOAD CURR SEG POISTION
065AR 4895                 LH     R9,6(R5)
        0006
065ER 2691                 AIS    R9,1
0660R D3B7                 LB     R11,1(R7)
        0001
0664R 059B                 CLHR   R9,R11             COMP CURR NO WITH TOTAL NO
0666R 2329                 BFFS   2,9
0668R 4200                 NOP
        0000
066CR               KILLS  EQU    *
                    *  SET END OF OBJ FLAG
066CR D215                 STB    R1,2(R5)
        0002
0670R D160                 LM     R6,RSAV1
        0708R
0674R 4300                 BFC    0,NXT1
        1E82R
                    *  STORE NEW POSITION
0678R 4095                 STH    R9,6(R5)
        0006
067CR CA79                 AHI    R7,1(R9)           INDEX NEW POSITION BY SEG POSITION
        0001
0680R D397                 LB     R9,0(R7)
        0000
0684R 4590                 CLH    R9,SEGNUM
        15FAR
0688R 4220                 BTC    2,KILLS
        066CR
068CR 0899                 LHR    R9,R9
068ER 4330                 BFC    3,KILLS
        066CR
                    *  STORF NEW NUMBER
0692R 4095                 STH    R9,X'24'(R5)
        0024
0696R 2791                 SIS    R9,1
0698R 0788                 XHR    R8,R8
069AR C8A0                 LHI    R10,X'18'
        0018
069ER 0C8A                 MHR    R8,R10
06A0R CA90                 AHI    R9,SEGLST          GENERATE SEGMENT INDEX
        114AR
06A4R 6809                 LE     R0,0(R9)
        0000
06A8R 4005                 STH    R0,X'2E'(R5)
        002E
06ACR 6809                 LE     R0,4(R9)
        0004
06B0R 6005                 STE    R0,X'1C'(R5)
        001C
                    *  SET REMAINING SEG DIST=SEG LEN
                    *  X AND Y COOR INITIALIZATION
06B4R 6809                 LE     R0,8(R9)
        0008
06B8R 6829                 LE     R2,12(R9)
```

```
      000C
06BCR 6849              LE     R4,X'10'(R9)
      0010
06C0R 6869              LE     R6,X'14'(R9)
      0014
06C4R D3A5              LB     R10,4(R5)
      0004
06C8R 08AA              LHR    R10,R10
06CAR 4330              BFC    3,NRVERS
      06EAR
06CER 6045              STE    R4,8(R5)
      0008
06D2R 6045              STE    R4,X'10'(R5)
      0010
06D6R 6065              STE    R6,12(R5)
      000C
06DAR 6065              STE    R6,X'14'(R5)
      0014
06DER 6005              STE    R0,X'26'(R5)
      0026
06E2R 6025              STE    R2,X'2A'(R5)
      002A
06E6R 4300              BFC    0,RET
      0702R
06EAR          NRVERS   EQU    *
06EAR 6005              STE    R0,8(R5)
      0008
06EER 6005              STE    R0,X'10'(R5)
      0010
06F2R 6025              STE    R2,12(R5)
      000C
06F6R 6025              STE    R2,X'14'(R5)
      0014
06FAR 6045              STE    R4,X'26'(R5)
      0026
06FER 6065              STE    R6,X'2A'(R5)
      002A
0702R          RET      EQU    *
0702R D160              LM     R6,RSAV1
      0708R
0706R 030E              BFCR   0,R14
0708R          RSAV1    DS     24
0720R          OBJLST   DS     2600
1148R          OBJNUM   DS     2
114AR          SEGLST   DS     1200
15FAR          SEGNUM   DS     2
15FCR          REGSAV   DS     32
161CR          FORSAV   DS     14
162AR 0000     OBJRD    DC     0,0
      0000
162ER 0000     CURGEO   DC     0
1630R 0000     CURSEN   DC     0
1632R 0000     CUROBJ   DC     0
1634R          ACTSEN   DS     1700
1CD8R          STRTBL   DS     2              STRING IDS + NUMBER OF SEGS
1CDAR                   DS     15             15 SEGMENTS PER STRING
```

```
1CEAR                  DS     408             25 STRINGS MAX
0011          STRLGT   EQU    17
1E82R         STRNO    DS     2
              * TASK CLOCK COUNT
1E84R 0000    CLK1     DC     0
              * TASK SECONDS COUNT
1E86R 0000    SEC1     DC     0
1E88R         SCNFLG   DS     2
1E8AR         SENSNO   DS     2
0022          @SENEN   EQU    X'22'
0000          @SEGM    EQU    0
0018          @SEGLT   EQU    X'18'
0000          @SENID   EQU    0
0032          @OBJMX   EQU    X'32'
0000          @OBJNO   EQU    0
0005          @OBJFG   EQU    5
0001          @STRNG   EQU    1
0002          @SENFG   EQU    2
0004          @SEGL    EQU    4
0003          @SENTP   EQU    3
0002          @OBJEN   EQU    2
0003          @OBJCL   EQU    3
0004          @SENDW   EQU    4
001E          @SENTK   EQU    X'1E'
0033          @OBJMY   EQU    X'33'
0014          @OBJIY   EQU    X'14'
0026          @OBJEX   EQU    X'26'
0004          @OBJD    EQU    4
0008          @SENRT   EQU    8
001F          @SENUK   EQU    X'1F'
0008          @SEG1X   EQU    8
000C          @SENSX   EQU    12
0020          @SENTB   EQU    X'20'
0034          @LSTL    EQU    X'34'
001C          @OBJRS   EQU    X'1C'
002E          @SEGSL   EQU    X'2E'
0008          @OBJCX   EQU    8
000C          @SEG1Y   EQU    12
0010          @SENSY   EQU    X'10'
001D          @SENW    EQU    X'1D'
0006          @SEGPO   EQU    6
0024          @SEGNO   EQU    X'24'
0010          @SEG2X   EQU    X'10'
0014          @SENPB   EQU    X'14'
0014          @SEG2Y   EQU    X'14'
0018          @SENDR   EQU    X'18'
000C          @OBJCY   EQU    12
0010          @OBJIX   EQU    X'10'
001C          @SENP    EQU    X'1C'
0018          @OBJVE   EQU    X'18'
0020          @OBJST   EQU    X'20'
002A          @OBJEY   EQU    X'2A'
1E8CR         SCENAR   EQU    *
1E8CR 0733             XHR    R3,R3           ZERO OBJECT NUMBER
1E8ER 4000             STH    R0,CLASFG
      0384R
```

```
1E92R 4840              LH     R4,OBJNUM
      1148R
1E96R          PROCES   EQU    *
1E96R 4000              STH    R0,SCNFLG
      1E88R
1E9AR 2421              LIS    R2,1
1E9CR 2431              LIS    R3,1
1E9ER C850              LHI    R5,OBJLST-@LSTL
      06ECR
1EA2R          NXTOBJ   EQU    *
1EA2R CA50              AHI    R5,X'34'
      0034
               * CHECK FOR END OF OBJECT
1EA6R D385              LB     R8,2(R5)
      0002
1EAAR 0888              LHR    R8,R8
1EACR 233C              BFFS   3,12
1EAER 4200              NOP
      0000
1EB2R          NXT1     EQU    *
1EB2R C120              BXLE   R2,NXTOBJ
      1EA2R
1EB6R 4820              LH     R2,SCNFLG
      1E88R
1EBAR 4330              BFC    3,ENDSCN
      24EER
1EBER 4000              STH    R0,SCNFLG
      1E88R
1EC2R 030D              BFCR   0,R13
1EC4R 4010     CONT1    STH    R1,SCNFLG
      1E88R
1EC8R 6800              LE     R0,SCETM          LOAD CURR SCEN TIME
      251ER
1ECCR 6905              CE     R0,X'20'(R5)
      0020
1ED0R 4280              BTC    8,NXT1
      1EB2R
1ED4R D385              LB     R8,5(R5)
      0005
1ED8R 0888              LHR    R8,R8
1EDAR 2339              BFFS   3,9
1EDCR 4200              NOP
      0000
               *  INITIALIZE CURR SEG POSITION
1EE0R 4005              STH    R0,6(R5)
      0006
1EE4R 41E0              BAL    R14,SETUP         SET UP WHEN OBJECT STARTS FOR F RS
      0644R
1EE8R D205              STB    R0,5(R5)
      0005
1EECR          CONT31   EQU    *
1EECR 6805              LE     R0,X'18'(R5)
      0018
1EF0R 6C00              ME     R0,DELTM
      251AR
1EF4R 6000              STE    R0,CURRD
```

```
        252AR
1EF8R 6000              STE    R0,OBJRD
        162AR
1EFCR            NXT2   EQU    *
1EFCR 6820              LE     R2,OBJRD
        162AR
1F00R 6805              LE     R0,X'10'(R5)
        0010
1F04R 2920              CER    R2,R0              COMP OBJ DIST WITH RFM SEG IST
1F06R 4230              BTC    2,NXTSEG
        2006R
1F0AR 2B02              SER    R0,R2
1F0CR 6005              STE    R0,X'10'(R5)
        0010
1F10R 6825              LE     R2,X'2E'(R5)
        002E
1F14R 2C22              MER    R2,R2
1F16R 6A20              AE     R2,C1
        038ER
1F1AR 6020              STE    R2,A               A=1+M**2
        252ER
1F1ER D0C0              STM    R12,REGSAV
        15FCR
1F22R D1C0              LM     R12,FORSAV
        161CR
1F26R 41F0              BAL    15,SQRT
        0000F
1F2AR 0004              DC     X'0004'
1F2CR 252ER             DC     A
1F2ER 6000              STE    0,B
        2532R
                 *  CONTINUE PROGRAM
1F32R D0C0              STM    R12,FORSAV
        161CR
1F36R D1C0              LM     R12,REGSAV
        15FCR
1F3AR 6820              LE     R2,CURRD
        252AR
1F3ER 6D20              DE     R2,B               R2=CURRD/B
        2532R
1F42R 6865              LE     R6,X'2E'(R5)
        002E
1F46R 6960              CE     R6,MAXM
        0364R
1F4AR 4320              BFC    2,NOVERT
        1F80R
1F4ER 6960              CE     R6,MINM
        0368R
1F52R 4280              BTC    8,NOVERT
        1F80R
1F56R 6820              LE     R2,CURRD
        252AR
1F5AR 6805              LE     R0,X'2A'(R5)
        002A
1F5ER 6B05              SE     R0,X'14'(R5)
        0014
```

```
                       * CHECK FOR MINUS CONDITION
1F62R 2319                   BFFS  1,9
1F64R 4200                   NOP
      0000
1F68R 48B0                   LH    R11,4
      0004
1F6CR C6B0                   OHI   R11,X'8000'
      8000
1F70R 40B0                   STH   R11,4
      0004
                       *  NOT MINUS CONDITION
1F74R 6A25                   AE    R2,12(R5)
      000C
1F78R 6025                   STE   R2,12(R5)
      000C
1F7CR 4300                   BFC   0,OBJMAP
      1FC6R
                       *  NO VERTICAL CALCULATION
1F80R                  NOVERT EQU   *
1F80R 6C65                   ME    R6,X'10'(R5)
      0010
1F84R 6845                   LE    R4,12(R5)
      000C
1F88R 6B45                   SE    R4,X'14'(R5)
      0014
1F8CR 2A64                   AER   R6,R4
1F8ER 6C65                   ME    R6,X'2E'(R5)
      002E
1F92R 6A65                   AE    R6,8(R5)
      0008
1F96R 6D60                   DE    R6,A
      252ER
1F9AR 6805                   LE    R0,X'26'(R5)
      0026
1F9ER 6B05                   SE    R0,X'10'(R5)
      0010
1FA2R 2317                   BFFS  1,7
1FA4R C8B0                   LHI   R11,4
      0004
1FA8R C6B0                   OHI   R11,X'8000'
      8000
1FACR 40B0                   STH   R11,4
      0004
1FB0R 2A62                   AER   R6,R2
1FB2R 6065                   STE   R6,8(R5)
      0008
1FB6R 6B65                   SE    R6,X'10'(R5)
      0010
1FBAR 6C65                   ME    R6,X'2E'(R5)
      002E
1FBER 6A65                   AE    R6,X'14'(R5)
      0014
1FC2R 6065                   STE   R6,12(R5)
      000C
                       * OBJECT MAP CALCULATIONS
1FC6R                  OBJMAP EQU   *
```

```
1FC6R 4880             LH     R8,MAPFLG
      034ER
1FCAR 4580             CLH    R8,MAPWT
      0350R
1FCER 4280             BTC    8,SENCAL
      20D4R
1FD2R 6805             LE     R0,8(R5)
      0008
1FD6R 6B00             SE     R0,SCXF
      032AR
1FDAR 4210             BTC    1,SENCAL
      20D4R
                 *  CONVERT TO BINARY
1FDER 41E0             BAL    R14,CNVBI
      229CR
1FE2R 034AR            DC     XLOC
1FE4R 4890             LH     R9,XLOC
      034AR
1FE8R 4210             BTC    1,SENCAL
      20D4R
                 *  CRT 1 CONSTANTS
1FECR 08B0             LHI    R11,CRT1DA
      03EAR
1FF0R 459B             CLH    R9,RX(R11)
      0004
1FF4R 4220             BTC    2,SENCAL
      20D4R
1FF8R 4B9B             SH     R9,XR(R11)
      0008
1FFCR 0788             XHR    R8,R8
1FFER 4D8B             DH     R8,INCX(R11)
      0006
2002R C790             XHI    R9,X'7F'
      007F
2006R C490             NHI    R9,X'7F'
      007F
200AR D290             STB    R9,XLOC
      034AR
200ER 6805             LE     R0,12(R5)
      000C
2012R 6B00             SE     R0,SCYF
      033AR
2016R 4210             BTC    1,SENCAL
      20D4R
                 *  CONVERT TO BINARY
201AR 41E0             BAL    R14,CNVBI
      229CR
201ER 034CR            DC     YLOC
2020R 4890             LH     R9,YLOC
      034CR
2024R 4210             BTC    1,SENCAL
      20D4R
2028R 459B             CLH    R9,RY(R11)
      000F
202CR 4220             BTC    2,SENCAL
      20D4R
```

```
                    *  Y CALCULATIONS
2030R 4B9B                    SH     R9,YR(R11)
      0012
2034R 0788                    XHR    R8,R8
2036R 4D8B                    DH     R8,INCY(R11)
      0010
203AR C880                    LHI    R8,19
      0013
203ER 0B89                    SHR    R8,R9
2040R C780                    XHI    R8,X'7F'
      007F
2044R C480                    NHI    R8,X'7F'
      007F
2048R D485                    CLB    R8,X'33'(R5)
      0033
204CR 4230                    BTC    3,TRYX
      206ER
2050R D3A0                    LB     R10,XLOC
      034AR
2054R D3C5                    LB     R12,X'32'(R5)
      0032
2058R 0BAC                    SHR    R10,R12
205AR C9A0                    CHI    R10,3
      0003
205ER 4230                    BTC    2,OUTLOC
      209AR
2062R C9A0                    CHI    R10,-3
      FFFD
2066R 4280                    BTC    8,OUTLOC
      209AR
206AR 4300                    BFC    0,SENCAL
      20D4R
                    *  TRY X CALCULATION
206ER D390          TRYX      LB     R9,XLOC
      034AR
2072R D495                    CLB    R9,X'32'(R5)
      0032
2076R 4230                    BTC    3,OUTLOC
      209AR
207AR 08A8                    LHR    R10,R8
207CR D3C5                    LB     R12,X'33'(R5)
      0033
2080R 0BAC                    SHR    R10,R12
2082R C9A0                    CHI    R10,1
      0001
2086R 212A                    BTFS   2,10
2088R 4200                    NOP
      0000
208CR C9A0                    CHI    R10,-1
      FFFF
2090R 2185                    BTFS   8,5
2092R 4200                    NOP
      0000
2096R 4300                    BFC    0,SENCAL
      20D4R
                    *  OUTPUT LOCATION
```

```
209AR                OUTLOC   EQU    *
209AR D285                    STB    R8,X'33'(R5)
      0033
209ER D280                    STB    R8,MAPBUF+2
      035ER
20A2R D380                    LB     R8,XLOC
      034AR
20A6R D285                    STB    R8,X'32'(R5)
      0032
20AAR D280                    STB    R8,MAPBUF+1
      035DR
20AER D385                    LB     R8,3(R5)
      0003
20B2R D388                    LB     R8,OBJTBL-1(R8)
      20C1R
20B6R D280                    STB    R8,MAPBUF+3
      035FR
20BAR E110                    SVC    1,OBJLOC
      0352R
20BER 4300                    BFC    0,SENCAL
      20D4R
20C2R 2E3A           OBJTBL   DC     X'2E3A',X'2200'
      2200
20C6R 2B20           NXTSEG   SER    R2,R0
20C8R 6020                    STE    R2,OBJRD          SAVE REMAINING DISTANCE
      162AR
20CCR 41E0                    BAL    R14,SETUP         SET UP NEW OBJECT LIST FOR NEXT SE
      0644R
20D0R 4300                    BFC    0,NXT2
      1EFCR
20D4R                SENCAL   EQU    *
20D4R C870                    LHI    R7,ACTSEN
      1634R
20D8R 0788                    XHR    R8,R8
20DAR 4890                    LH     R9,SENSNO
      1E8AR
20DER 2791                    SIS    R9,1
20E0R C8A0                    LHI    R10,X'22'
      0022
20E4R 0C8A                    MHR    R8,R10
20E6R 088A                    LHR    R8,R10
20E8R 0A97                    AHR    R9,R7
20EAR                NXTSEN   EQU    *
20EAR D3A7                    LB     R10,2(R7)
      0002
20EER 08AA                    LHR    R10,R10
20F0R 4230                    BTC    3,NXTVAL
      2206R
20F4R                CONT7    EQU    *
20F4R 6807                    LE     R0,X'18'(R7)
      0018
20F8R D3A5                    LB     R10,3(R5)
      0003
20FCR C5A0                    CLHI   R10,1             IS OBJECT PERSONNEL?
      0001
2100R 2134                    BTFS   3,4
```

```
2102R 6800              LE     R0,C50
      038AR
2106R 2306              BFFS   0,6
2108R C5A0              CLHI   R10,2            IS OBJECT WHEEL
      0002
210CR 2133              BTFS   3,3
210ER 6C00              ME     R0,COF6
      0376R
2112R 6885              LE     R8,8(R5)
      0008
2116R 6B87              SE     R8,12(R7)
      000C
211AR 48B0              LH     R11,X'10'
      0010
211ER C4B0              NHI    R11,X'7FFF'      ABSOLUTE VAL OF FP R8
      7FFF
2122R 40B0              STH    R11,X'10'
      0010
2126R 2980              CER    R8,R0
2128R 4220              BTC    2,NXTVAL
      2206R
212CR 68A5              LE     R10,12(R5)
      000C
2130R 6BA7              SE     R10,16(R7)
      0010
2134R 48B0              LH     R11,X'14'
      0014
2138R C4B0              NHI    R11,X'7FFF'
      7FFF
213CR 40B0              STH    R11,X'14'
      0014
2140R 29A0              CER    R10,R0
2142R 4220              BTC    2,NXTVAL
      2206R
                  *  SAVE DETECTION RADIUS
2146R 6000              STE    R0,DETRAD
      2546R
214AR 40A0              STH    R10,OBJTP
      220ER
214ER 2C88              MER    R8,R8
2150R 2CAA              MER    R10,R10
2152R 2A8A              AER    R8,R10
2154R 6080              STE    R8,W
      24FER
2158R D3A7              LB     R10,3(R7)
      0003
215CR 08AA              LHR    R10,R10
215ER 2333              BFFS   3,3
2160R 4010              STH    R1,CLASFG
      0384R
                  *  SET UP CALCULATION
2164R             CONT9 EQU    *
2164R D0C0              STM    R12,REGSAV
      15FCR
2168R D1C0              LM     R12,FORSAV
      161CR
```

```
216CR 41F0               BAL    15,SORT
      1F28R
2170R 0004               DC     X'0004'
2172R 24FER              DC     W
2174R 6000               STE    0,SENOBD
      254AR
2178R D0C0               STM    R12,FORSAV
      161CR
217CR D1C0               LM     R12,REGSAV
      15FCR
                *  CHECK SENSOR TO OBJECT DISTANCE
2180R 6860               LE     R6,DETRAD
      2546R
2184R 6960               CE     R6,SENOBD
      254AR
2188R 4280               BTC    8,NXTVAL
      2206R
218CR 6800               LE     R0,C1
      038ER
                *  CHECK PROBABILITY THRESHOLD
2190R 6900               CE     R0,PRGATE
      0394R
2194R 4280               BTC    8,NXTVAL
      2206R
2198R 6827               LE     R2,X'14'(R7)
      0014
219CR 2A20               AER    R2,R0
219ER 6027               STE    R2,X'14'(R7)
      0014
                *  CHECK TYPE
21A2R 48A0               LH     R10,CLASFG
      0384R
21A6R 4330               BFC    3,NXTVAL
      2206R
21AAR 4000               STH    R0,CLASFG
      0384R
21AER 07AA               XHR    R10,R10
21B0R 48B0               LH     R11,II
      0382R
21B4R 4CA0               MH     R10,C899
      0388R
21B8R C4B0               NHI    R11,X'7FFF'
      7FFF
21BCR 40B0               STH    R11,II
      0382R
21C0R 45B0               CLH    R11,GATE
      0386R
21C4R 212A               BTFS   2,10
21C6R 4200               NOP
      0000
21CAR C8A0               LHI    R10,X'1B'
      001B
21CER 4AA0               AH     R10,OBJTP          SET CORRECT CLASSIFICATION
      2206R
21D2R 0AA7               AHR    R10,R7
21D4R 4300               BFC    0,STOREC
```

```
      21FCR
21D8R 48B0     FALSEC  LH     R11,OBJTP
      220ER
21DCR 27B1             SIS    R11,1
21DER 07AA             XHR    R10,R10
21E0R 4CA0             MH     R10,LUS
      0018R
21E4R 48A0             LH     R10,II
      0382R
21E8R C4A0             NHI    R10,2
      0002
21ECR 90A1             SRLS   R10,1
21EER 0AAB             AHR    R10,R11
21F0R D3CA             LB     R12,FALTBL(R10) FETCH FALSE CLASSIFICATION
      2210R
21F4R C8A0             LHI    R10,X'1B'
      001B
21F8R 0AAC             AHR    R10,R12
21FAR 0AA7             AHR    R10,R7
21FCR D33A     STOREC  LB     R3,0(R10)
      0000
2200R 2631             AIS    R3,1
2202R D23A             STB    R3,0(R10)        STORE NEW COUNT
      0000
2206R C170     NXTVAL  BXLE   R7,NXTSEN
      20EAR
220AR 4300             BFC    0,NXT1
      1EB2R
220ER          OBJTP   DS     2                OBJECT TYPE
2210R 0203     FALTBL  DC     X'203',X'103',X'102'
      0103
      0102
2216R          CNVFP   EQU    *
2216R D060             STM    R6,SUBSAV
      2284R
221AR 2467             LIS    R6,7
221CR 488E             LH     R8,0(R14)
      0000
2220R 4888             LH     R8,0(R8)
      0000
2224R 489E             LH     R9,2(R14)
      0002
2228R 4899             LH     R9,0(R9)
      0000
               *  SET UP
222CR C480             NHI    R8,X'FF'
      00FF
2230R 07AA             XHR    R10,R10
2232R 07BB             XHR    R11,R11
               *  NEXT TRY
2234R 08C8     NXTTRY  LHR    R12,R8
2236R 2761             SIS    R6,1
2238R 4330             BFC    3,ERROR
      2268R
223CR 90C4             SRLS   R12,4
223ER 08CC             LHR    R12,R12
```

```
2240R 4230              BTC    3,OREXIT
      224CR
2244R EB80              RLL    R8,4
      0004
2248R 4300              BFC    0,NXTTRY
      2234R
              *  OR EXIT PATH
224CR         OREXIT    EQU    *
224CR C660              OHI    R6,X'40'
      0040
2250R 9466              EXBR   R6,R6
2252R 06A6              OHR    R10,R6
2254R 06A8              OHR    R10,R8
2256R 06B9              OHR    R11,R9
2258R 40A0              STH    R10,X'10'
      0010
225CR 40B0              STH    R11,X'12'
      0012
2260R D160              LM     R6,SUBSAV
      2284R
2264R 430E              BFC    0,4(R14)
      0004
              *  ERROR ROUTINE
2268R E120    ERROR     SVC    2,CERR
      2270R
226CR 4300              BFC    0,EOJ1
      24F2R
2270R 0007    CERR      DC     7,16,C'CONVERSION ERROR'
      0010
      434F
      4E56
      4552
      5349
      4F4E
      2045
      5252
      4F52
2284R         SUBSAV    DS     24
229CR         CNVBI     EQU    *
229CR D060              STM    R6,SUB2S
      2392R
22A0R 0700              XHR    R0,R0
22A2R 07CC              XHR    R12,R12
22A4R D3A0              LB     R10,0
      0000
22A8R 25D1              LCS    R13,1
22AAR C4A0              NHI    R10,X'F'
      000F
22AER 40A0              STH    R10,EXP
      2390R
22B2R D3B0              LB     R11,1
      0001
22B6R 90B4              SRLS   R11,4
              *  FACTOR 1
22B8R 41F0              BAL    R15,CALFA
      235CR
```

```
22BCR 07AA              XHR    R10,R10
22BER 4CA0              MH     R10,FACTOR
      238ER
22C2R 0ACB              AHR    R12,R11
22C4R 61D0              AHM    R13,EXP
      2390R
22C8R 4330              BFC    3,EXIT55
      234CR
22CCR D3B0              LB     R11,1
      0001
22D0R C4B0              NHI    R11,X'F'
      000F
                  *  FACTOR 2
22D4R 41F0              BAL    R15,CALFA
      235CR
22D8R 07AA              XHR    R10,R10
22DAR 4CA0              MH     R10,FACTOR
      238ER
22DER 0ACB              AHR    R12,R11
22E0R 61D0              AHM    R13,EXP
      2390R
22E4R 4330              BFC    3,EXIT55
      234CR
22E8R D3B0              LB     R11,2
      0002
22ECR 90B4              SRLS   R11,4
                  *  FACTOR 3
22EER 41F0              BAL    R15,CALFA
      235CR
22F2R 07AA              XHR    R10,R10
22F4R 4CA0              MH     R10,FACTOR
      238ER
22F8R 0ACB              AHR    R12,R11
22FAR 61D0              AHM    R13,EXP
      2390R
22FER 4330              BFC    3,EXIT55
      234CR
2302R D3B0              LB     R11,2
      0002
2306R C4B0              NHI    R11,X'F'
      000F
                  *  FACTOR 4
230AR 41F0              BAL    R15,CALFA
      235CR
230ER 07AA              XHR    R10,R10
2310R 4CA0              MH     R10,FACTOR
      238ER
2314R 0ACB              AHR    R12,R11
2316R 61D0              AHM    R13,EXP
      2390R
231AR 4330              BFC    3,EXIT55
      234CR
231ER D3B0              LB     R11,3
      0003
2322R 90B4              SRLS   R11,4
                  *  FACTOR 5
```

```
2324R 41F0                 BAL    R15,CALFA
      235CR
2328R 07AA                 XHR    R10,R10
232AR 4CA0                 MH     R10,FACTOR
      238ER
232ER 0ACB                 AHR    R12,R11
2330R 61D0                 AHM    R13,EXP
      2390R
2334R 4330                 BFC    3,EXIT55
      234CR
2338R D3B0                 LB     R11,3
      0003
233CR C4B0                 NHI    R11,X'F'
      000F
2340R 41F0                 BAL    R15,CALFA
      235CR
2344R 07AA                 XHR    R10,R10
2346R 4CA0                 MH     R10,FACTOR
      238ER
234AR 0ACB                 AHR    R12,R11
                 *  EXIT SUBPROGRAM
234CR            EXIT55    EQU    *
234CR 48BE                 LH     R11,0(R14)
      0000
2350R 40CB                 STH    R12,0(R11)
      0000
2354R D160                 LM     R6,SUB25
      2392R
2358R 430E                 BFC    0,2(R14)
      0002
                 *  CALCULATE FACTOR
235CR            CALFA     EQU    *
235CR 2411                 LIS    R1,1
235ER 4010                 STH    R1,FACTOR
      238ER
2362R 4880                 LH     R8,EXP
      2390R
2366R 2781                 SIS    R8,1
2368R 033F                 BFCR   3,R15
236AR C870                 LHI    R7,16
      0010
236ER 2781                 SIS    R8,1
2370R 233B                 BFFS   3,11
2372R 4200                 NOP
      0000
2376R            NEXT16    EQU    *
2376R 0766                 XHR    R6,R6
2378R 4C60                 MH     R6,SIXTNN
      238CR
237CR 2781                 SIS    R8,1
237ER 2333                 BFFS   3,3
2380R 2205                 BFBS   0,5
2382R 4200                 NOP
      0000
2386R            NEXT17    EQU    *
2386R 4070                 STH    R7,FACTOR
```

```
        238ER
238AR 030F            BFCR   0,R15
238CR 0010   SIXTNN   DC     16
238ER        FACTOR   DS     2
2390R        EXP      DS     2
2392R        SUB2S    DS     24
000B         WORK1    EQU    11
000C         WORK     EQU    12
23AAR        ADDLST   EQU    *
23AAR D000            STM    R0,REGSAV
      15FCR
23AER D3C3            LB     WORK,0(R3)
      0000
23B2R 40C0            STH    WORK,LU
      2484R
             * LOAD ID VALUE IN HEX
23B6R D3C3            LB     WORK,1(R3)
      0001
23BAR 41A0            BAL    R10,INDEX
      248AR
23BER D383            LB     R8,3(R3)
      0003
23C2R 0888            LHR    R8,R8
23C4R 2136            BTFS   3,6
23C6R 4200            NOP
      0000
23CAR 0788            XHR    R8,R8
23CCR 4300            BFC    0,ADDMSG
      23FCR
             * CHECK FOR DATA TYPE
23D0R D383            LB     R8,X'1E'(R3)
      001E
23D4R 0888            LHR    R8,R8
23D6R 2334            BFFS   3,4
23D8R 2483            LIS    R8,3
23DAR 4300            BFC    0,EEND
      23F8R
23DER D383            LB     R8,X'1D'(R3)
      001D
23E2R 0888            LHR    R8,R8
23E4R 2333            BFFS   3,3
23E6R 2482            LIS    R8,2
23E8R 2308            BFFS   0,8
23EAR D383            LB     R8,X'1C'(R3)
      001C
23EER 0888            LHR    R8,R8
23F0R 2333            BFFS   3,3
23F2R 2481            LIS    R8,1
23F4R 2302            BFFS   0,2
23F6R 0788            XHR    R8,R8
23F8R        EEND     EQU    *
23F8R CA80            AHI    R8,52           ADJUST DATA
      0034
23FCR 94CC   ADDMSG   EXBR   WORK,WORK       SHIFT DATA
23FER 06C8            OHR    WORK,R8
2400R 91C2            SLLS   WORK,2
```

```
2402R 4000              STH    WORK,DATA        SAVE DATA
      2486R
2406R 4890              LH     R9,TIME
      247CR
240AR 9389              LBR    R8,R9
240CR C490              NHI    R9,X'F00'
      0F00
2410R 9194              SLLS   R9,4
2412R C480              NHI    R8,X'F'
      000F
2416R 9488              EXBR   R8,R8
2418R 0698              OHR    R9,R8            1ST BYTE R9 HAS HR
241AR D380              LB     R8,TIME+4
      2480R
241ER C480              NHI    R8,X'F'
      000F
2422R 0698              OHR    R9,R8            LS BIT MIN SET
2424R D380              LB     R8,TIME+3
      247FR
2428R C480              NHI    R8,X'F'
      000F
242CR 9184              SLLS   R8,4
242ER 0698              OHR    R9,R8            MS BIT MIN SET
2430R 4090              STH    R9,TIME1         SAVE MS BITS OF TIME
      2488R
2434R 4890              LH     R9,TIME+6
      2482R
2438R 9389              LBR    R8,R9
243AR C490              NHI    R9,X'F00'
      0F00
243ER 9194              SLLS   R9,4
2440R C480              NHI    R8,X'F'
      000F
2444R 9488              EXBR   R8,R8
2446R 0698              OHR    R9,R8
2448R 4870              LH     R7,LU            GET LU
      2484R
244CR 4880              LH     R8,DATA          GET DATA
      2486R
2450R 4800              LH     WORK,TIME1       GET TIME1
      2488R
2454R 6570              ABL    R7,INFO
      0000F
2458R 6580              ABL    R8,INFO
      2456R
245CR 6500              ABL    WORK,INFO
      245AR
2460R 6590              ABL    R9,INFO          ADD SECOO
      245ER
2464R 2411              LIS    R1,1
                  *  SET SENSOR FLAG
2466R D213              STB    R1,2(R3)
      0002
246AR D100              LM     R0,REGSAV
      15FCR
246ER 6803              LE     R0,4(R3)
```

```
        0004
2472R  6003              STE    R0,8(R3)
        0008
2476R  030D              BFCR   0,R13
2478R  0008    ROTIME    DC     8,TIME
        247CR
247CR          TIME      DS     8
2484R          LU        DS     2
2486R          DATA      DS     2
2488R          TIME1     DS     2
248AR          INDEX     EQU    *
248AR  2701              SIS    WORK,1            ID-1
248CR  C8CC              LHI    WORK,CNVTAB(WORK)
        2496R
2490R  D3CC              LB     WORK,0(WORK)      FETCH INDEX
        0000
2494R  030A              BFCR   0,R10
2496R          CNVTAB    EQU    *
2496R  1516              DC     X'1516'
2498R  1719              DC     X'1719'
249AR  1A1B              DC     X'1A1B'
249CR  1D1E              DC     X'1D1E'
249ER  1FFF              DC     X'1FFF'
24A0R  2526              DC     X'2526'
24A2R  2729              DC     X'2729'
24A4R  2A2B              DC     X'2A2B'
24A6R  2D2E              DC     X'2D2E'
24A8R  2FFF              DC     X'2FFF'
24AAR  3536              DC     X'3536'
24ACR  3739              DC     X'3739'
24AER  3A3B              DC     X'3A3B'
24B0R  3D3E              DC     X'3D3E'
24B2R  3FFF              DC     X'3FFF'
24B4R  0506              DC     X'506'
24B6R  0709              DC     X'709'
24B8R  0A0B              DC     X'A0B'
24BAR  0D0E              DC     X'D0E'
24BCR  0FFF              DC     X'FFF'
24BER  1112              DC     X'1112'
24C0R  1321              DC     X'1321'
24C2R  2223              DC     X'2223'
24C4R  3132              DC     X'3132'
24C6R  33FF              DC     X'33FF'
24C8R  1418              DC     X'1418'
24CAR  1C24              DC     X'1C24'
24CCR  282C              DC     X'282C'
24CER  3438              DC     X'3438'
24D0R  3CFF              DC     X'3CFF'
24D2R  0102              DC     X'102'
24D4R  0304              DC     X'304'
24D6R  080C              DC     X'80C'
24D8R  1020              DC     X'1020'
24DAR  30FF              DC     X'30FF'
24DCR  0000              DC     0
24DER  0007    ENSCEN    DC     7,12,C'END SCENARIO'
        000C
```

```
         454E
         4420
         5343
         454E
         4152
         494F
24EER  E120     ENDSCN   SVC    2,ENSCEN
         240ER
24F2R           EQU1     EQU    *
24F2R  D100              LM     RO,FORSAV
         161CR
24F6R                    EXTRN  .V
24F6R  41F0              BAL    15,.V
         0000F
24FAR           DELTIM   DS     2
24FCR           SCETIM   DS     2
24FER           W        DS     4
2502R           X        DS     4
2506R           Y        DS     4
250AR           INITSX   DS     4
250ER           INITSY   DS     4
2512R           CURRX    DS     4
2516R           CURRY    DS     4
251AR           DELTM    DS     4
251ER           SCETM    DS     4
2522R           ENDX     DS     4
2526R           ENDY     DS     4
252AR           CURRD    DS     4
252ER           A        DS     4
2532R           B        DS     4
2536R           C        DS     4
253AR           D        DS     4
253ER           SENSX    DS     4
2542R           SENSY    DS     4
2546R           DETRAD   DS     4
254AR           SENOBD   DS     4
254ER           ACTPRB   DS     4
2552R           CURRM    DS     4
2556R  4278     $C2      DC     X'4278',X'0000'
         0000
255AR                    END
```

☆ U.S. GOVERNMENT PRINTING OFFICE: 1981 – 703-073/FM-C-48-81

HISA-

DATE

ILME